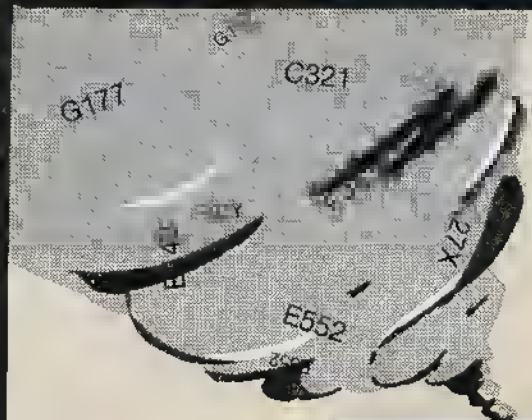


AIM SYM KIM APPLE ATARI PET OSI AIM SYM KIM APPLE ATARI APPLE KIM ATARI OSI AIM PET APPLE SYM KIM ATARI
 T AIM SYM KIM ATARI APPLE PET OSI AIM SYM KIM ATARI APPLE PET OSI APPLE PET OSI AIM SYM KIM ATARI PET APPI
 T APPLE ATARI KIM SYM AIM PET ATARI APPLE OSI AIM SYM KIM APPLE ATARI PET OSI AIM SYM KIM ATARI APPLE PE
 T SYM KIM AIM ATARI OSI PET AIM SYM KIM ATARI PET OSI AIM SYM ATARI PET OSI APPLE PET OSI
 PLE ATARI KIM SYM OSI PET SIM KIM KIM SYM ATARI OSI SYM KIM AIM KIM PET A
 I PET AIM SYM ATARI KIM PET OSI A KIM SYM AIM PET ATARI APPLE OS
 A APPLE ATARI PET OSI AIM SYM KIM A ATARI OSI AIM PET APPLE SYM KIM PET ATARI APPLE
 T OSI PET AIM SYM KIM APPLE PET OSI PET SYM KIM ATARI OSI PET AIM OSI APPLE PET OSI SYM KIM AIM PET APPLE OS
 A PET APPLE PET SYM KIM APPLE PET O PET AIM KIM ATARI OSI PET AIM SYM KIM PET APPLE ATARI PE
 A PET SIM KIM A PET APPLE PET A PET KIM PET PET KIM PET PET AIM SYM KIM A
 ARI SYM KIM APPLE PET ATARI SYM KIM PET PET PET PET PET PET PET PET PET OSI PET ATAF
 A SYM OSI OSI APPLE PET AIM SYM PET APPLE ATARI SYM KIM OSI APPLE PET APPLE KIM SYM APPLE PET OSI PET ATAF
 M KIM ATARI PETKIM PET APPLE ATARI OSI PET SYM KIM ATARI PET APPLE ATARI OSI PET SYM KIM ATARI PET APPLE

MICRO

6502

JOURNAL



PET Keysort

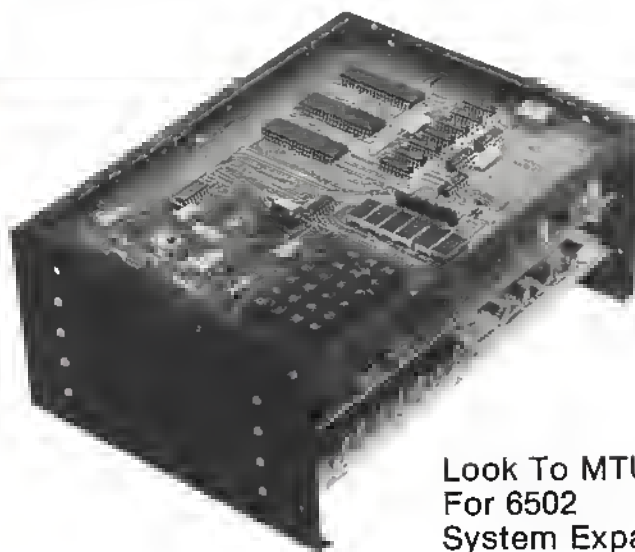


V KIM ATARI APPLE PET OSI AIM SYM KIM ATARI APPLE PET OSI OSI PET APPLE ATARI KIM SYM AIM PET ATARI APP
 I SYM KIM APPLE ATARI PET OSI AIM SYM KIM APPLE ATARI APPLE KIM ATARI OSI AIM PET APPLE SYM KIM ATARI A
 I SYM KIM ATARI APPLE PET OSI AIM SYM KIM ATARI APPLE PET OSI APPLE PET OSI AIM SYM KIM ATARI PET APPLE C
 PLE ATARI KIM SYM AIM PET ATARI APPLE OSI AIM SYM KIM APPLE ATARI PET OSI AIM SYM KIM ATARI APPLE PET C
 M KIM AIM ATARI OSI PET AIM SYM KIM PET OSI PET APPLE ATARI PET OSI AIM SYM KIM ATARI PET OSI APPLE PET OSI F
 ATARI KIM SYM OSI PET SIM KIM ATARI PET A PET OSI SYM KIM AIM PET APPLE ATARI OSI SYM KIM AIM KIM PET APP
 AIM SYM ATARI KIM PET OSI APPLE ATARI APPLE PET OSI PET APPLE ATARI KIM SYM AIM PET ATARI APPLE OSI S
 PLE ATARI PET OSI AIM SYM KIM APPLE ATARI PET SYM KIM ATARI OSI AIM PET APPLE SYM KIM PET ATARI APPLE AT
 I PET AIM SYM KIM APPLE PET OSI PET SYM KIM ATARI OSI PET AIM OSI APPLE PET OSI SYM KIM AIM PET APPLE OSI F

NO. 23 APRIL 1980 \$2.00



PET-AIM KIM-SYM



Look To MTU
For 6502
System Expansion

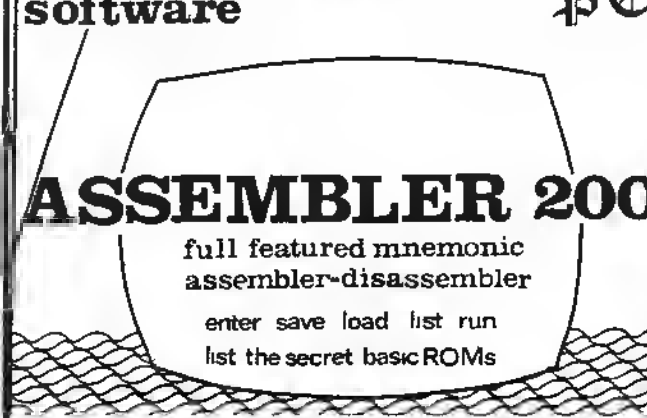



Micro Technology Unlimited

P.O. Box 4596, 841 Galaxy Way
Manchester, N.H. 03108

603-627-1464

Call Or Write For Our Full Line Catalog

<p>softside software \$15.95 PET</p>  <p>ASSEMBLER 2001 full featured mnemonic assembler-disassembler enter save load list run list the secret basic ROMs</p>	<p>softside software \$9.95 PET</p>  <p>DRIVING ACE ★ ★ 2 ACTION PACKED VIDEO GAMES ★ ★</p>
--	--

PET BASIC BREAKTHROUGH

Softside Software

presents

SYMBOLIC/STRUCTURED BASIC

At last, **Symbolic/Structured Basic** is available for your PET 8-32K personal computer! S-Basic is a pre-compiler that enhances the PET's built-in basic monitor with the addition of extra-control statements found only in the most sophisticated computers. **WHILE ... GOSUB ...** calls a subroutine as long as a condition is true. **UNTIL ... GOSUB ...** jumps to a subroutine unless the condition is true. The **IF ... Then ... ELSE** statement allows the programmer to command the computer to execute instructions if the normal IF condition is not met.

Forget about line numbers, S-Basic allows you to program naturally only naming (numerically or alphabetically!) statements that you will need to refer to, for ex-

ample: **LOOP/PRINT "HI": GO TO LOOP**. S-Basic program lines can be up to 255 characters long, two-and-one-half times as long as on standard Basic. S-Basic does not compromise any of PET Basic's existing features. All PET Basic commands can be used.

S-Basic includes an **editor** with full text capabilities, a **translator/pre-compiler** with its own error messages, and the **S-Basic loader**. These programs are recommended for disk-based PETs. A printer is optional but suggested. Cassette copies are available and require two cassette drives. Comprehensive instructions are included. Symbolic-/Structured Basic package is available complete for an **introductory price of \$35.95**.

- WHILE....GOTO....
- IF....THEN....ELSE
- 255 CHAR. LINES

**A PET
PROGRAMMING
BREAKTHROUGH**

- UNTIL....GOTO....
- SYMBOLIC OPTIONAL
- LINE NUMBERS

305 RIVERSIDE DRIVE, NEW YORK, NEW YORK 10025

PERFECT AIM



ATTRACTIVE FUNCTIONAL PACKAGING FOR YOUR AIM-65 MICROCOMPUTER

- Professional Appearance
- Striking Grey and Black Color Combination
- Protects Vital Components

ENGINEERED SPECIFICALLY FOR THE ROCKWELL AIM-65

- All Switches Accessible
- Integral Reset Button Actuator
- Easy Paper Tape Replacement

EASILY ASSEMBLED

- Absolutely No Alteration of AIM-65 Required
- All Fasteners Provided
- Goes Together in Minutes

MADE OF HIGH IMPACT STRENGTH THERMOFORMED PLASTIC

- Kydex 100*
- Durable
- Molded-In Color
- Non-Conductive

AVAILABLE FROM STOCK

- Allow Three to Four Weeks for Processing and Delivery
- No COD's Please
- Dealer Inquiries Invited

TO ORDER: 1. Fill in this Coupon (Print or Type Please)
2. Attach Check or Money Order and Mail to:

NAME _____

STREET _____

CITY _____

STATE _____ ZIP _____

SAE 1-1 PLEASE SHIP PREPAID _____ SAE 1-1(s)
@ \$43.50 each
California Residents Please Pay
\$46.33 (Includes Sales Tax)

SAE 1-2 PLEASE SHIP PREPAID _____ SAE 1-2(s)
@ \$46.50 each
California Residents Please Pay
\$49.52 (Includes Sales Tax)

**enclosures
group**

771 bush street
san francisco, california 94108

*TM Rohm & Hass Patent Applied For

Table of Contents

Applesoft II Shorthand by Allen J. Lacy	5
Editorial	9
The APPLE Stripper by Bill Crouch	11
Graphics and the Challenger C1P, Part 4 by William L. Taylor	15
SYMple BASIC Data Files John M. Blalock	21
A Perpetual Calendar Printer for the AIM by Mel Evans	27
Bi-directional Scrolling by Roger Wagner	31
The SY6516 Pseudo-16 Bit Processor by Randall Hyde	36
PET Keysort by Rev. James Strasma	43
KIM Scorekeeper by Joel Swank	59
OSI BASIC in ROM — What's Where? by Earl D. Morris, Jr.	65
Editorial	67
The MICRO Software Catalogue: XIX by Mike Rowe	71
6502 Bibliography: Part XIX by William R. Dial	77

Staff

Editor/Publisher
Robert M. Tripp
Associate Editor
Mary Ann Curtis
Business Manager
Maggie E. Fisher
Circulation Manager
Carol A. Stark
Art/Advertising Coordinator
Terry Spillane
Comptroller
Donna M. Tripp
Production Assistant
L. Catherine Bland
MICRO Lab
R. Keith Beal

MICRO™ is published monthly by MICRO INK, Inc., Chelmsford, MA 01824. Tel. 617/256-5515.
Second Class postage paid at Chelmsford, MA 01824.
Publication Number: COTR 395770.
Circulation: Paid subscriptions: U.S.: 3800, Foreign: 350; Dealers: U.S.: 4500, Foreign: 1900.
Subscription rates: U.S.: \$15 per year. Foreign, surface mail: \$18 per year.
For air mail rate, change of address, back issue or subscription information write to: MICRO, P.O. Box 6902, Chelmsford, MA 01824.
Entire contents Copyright © 1980 by MICRO INK, Inc.

Advertiser's Index

Aardvark	79	Instant Software, Inc.	74,75
AB Computers	56	Micro Technology Unlimited	IFC,26
Andromeda Computer Systems	22	NIBBLE	13
Automated Simulation	34	Powersoft, Inc.	10
Avante Guard Creations	19	Programma International	35
Beta Computer Devices	63	Progressive Computing	19
CJM-Industries	68	Progressive Computer Software	56
Classified Ads	8,12,45	Progressive Software	4
Computer Applications Tomorrow	79	Rainbow Computing	IBC
The Computerist, Inc.	57	RAYGAM	73
Computer Shop	62	RNB-Enterprises	20
Connecticut microComputers	40,41,70,79	Shepardson Microsystems	39
Creative Computing	64	SKYLES Electric Works	42,67
Cyberdyne	29	Small Business Computer Sys.	63
Dakin 5	38	Softape	BC
Decision Systems	56	Softside Publications	30
Digital Engineering Assoc.	80	Softside Software	1
Discount Data Products	80	Southeastern Software	58
Electronic Specialists, Inc.	38	Southwestern Data Systems	70
Enclosures Group	2	Stoneware	70
Galaxy	80	Synergistic Software	33
Hudson Digital Electronics	76	United Software of America	38
Information Unlimited Software	14		

PROGRESSIVE SOFTWARE

Presents
Software and Hardware for your APPLE

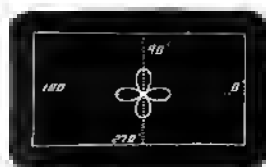
Missile—Anti—Missile (Aplstt)

\$9.95



Polar Coordinator Plot

\$9.95



by TD Moteles

Tables Generator forms shape tables with ease from directional vectors—starting address, length and position of each shape. Program saves shape tables anywhere in usable memory.

Murray Summers

\$9.95

Sales Forecast provides the best forecast using the four most popular forecasting techniques: linear regression, log trend, power curve trend, and exponential smoothing. Neil D. Lipson

\$9.95

Single Drive Copy is a special utility program, written by Vince Corsetti in Integer BASIC, that will copy a diskette using only one drive. It is supplied on tape and should be loaded on to a diskette. It automatically adjusts for APPLE memory size and should be used with DOS 3.2

\$19.95

Curve Fit accepts any number of data points, distributed in any fashion, and fits a curve to the set of points using log curve fit, exponential curve fit, least squares, or a power curve fit. It will compute the best fit, or employ a specific type of fit, and display a graph of the result. By Dave Garson.

\$9.95

Touch Typing Tutor teaches typing. Indicates speed and errors made. Finger Bldrs, Gen. Typing, Basic Language and User Supplied. Diskette. Written by Wm. A. Massena.

\$19.95

Apple Menu Cookbook index-accessed data storage/retrieval program. Recipes stored, unlimited lines per entry. Easy editing. Formulated after N.Y. Times Cookbook. Other useful features included. Written by Wm. Merlino, M.D.

\$19.95

Mailing List Program maintains complete record of name, address, phone no., mailing labels accommodated parallel card or built-in printer driver, easy data entry Diskette. 32K.

\$19.95

POSTAGE AND HANDLING

Please add \$1.25 for the first item and \$.75 for each additional item.

- Programs accepted for publication
- Highest royalty paid

Utility Pack combines four versatile programs by Vince Corsetti, for any memory configuration.

• **Integer to Applesoft conversion:** Encounter only those syntax errors unique to Applesoft after using this program to convert any Integer BASIC source.

• **Disk Append:** Merge any two Integer BASIC sources into a single program on disk.

• **Integer BASIC copy:** Replicate an Integer BASIC program from one disk to another, as often as required, with a single keystroke.

• **Applesoft Update:** Modify Applesoft on the disk to eliminate the heading always produced when it is first run.

• **Binary Copy:** Automatically determines the length and starting address of a program while copying its binary file from one disk to another in response to a single keystroke. \$9.95

Solitaire — Old European peg game, played by one (similar to Chinese checkers). Object — to finish with last peg in center.

Charles B. Smith

\$9.95

Water The Flowers — Math (add., sub., mult., div.,) grades 1-6. (disk)

Judy Pegg

\$9.95

Othello one or two players similar to chess in strategy. You can win with a single move. Vince Corsetti's program keeps board details and flip piece.

\$19.95

Blockade lets two players compete by building walls to obstruct each other. An exciting game written in Integer BASIC by Vince Corsetti.

\$9.95

Saucer Invasion, Space Maze, Starwars, Rocket Pilot:

Written by Bob Bishop

Each \$9.95

Hardware

Light Pen with seven supporting routines. The light meter takes intensity readings every fraction of a second from 0 to 588. The light graph generates a display of light intensity on the screen. The light pen connects points that have been drawn on the screen, in low or high resolution, and displays their coordinates. A special utility displays any number of points on the screen, for use in menu selection or games, and selects a point when the light pen touches it. The package includes a light pen calculator and light pen TIC TAC TOE. Neil D. Lipson's programs use artificial intelligence and are not confused by outside light. The HIRFS light pen, only requires 48K and ROM card

\$34.95

TO ORDER

Send Check or Money Order to:

P.O. Box 273

Plymouth Meeting, PA 19462

PA residents add 6% sales tax.

U.S. and foreign dealer and distributor inquiries invited
All programs require 16K memory unless specified

Applesoft II Shorthand

If you want to make Applesoft a little easier to use, try this program which permits entire commands to be input with a single control key. Since the command lookup is table driven, you can select the keys to conform to your own preferences. The techniques used provide a valuable understanding of how to add your own modifications.

Allen J. Lacy
1921 W. Oglethorpe
Albany, GA 31707

This routine allows a programmer to type in an entire Applesoft command with the use of one control key.

Overview

The routine Shorthand ties into the input hooks at \$38 and \$39 (56 and 57 decimal) and uses a table inside the RAM version of Applesoft II. In Applesoft's table, each command is represented as an ASCII string with the high bit off except for the last character of the string which has the high bit set. The routine also uses a monitor routine to read a key. If it is a control character, shorthand gets an address from its internal table. If the high byte of the address is 0, the routine passes the control character back. If the address is not 0 shorthand passes the command stored at that location back.

Step 1 turns DOS off. Step 2 turns Shorthand on. Step 3 turns DOS back on. But DOS will not be on at the same time as shorthand.

To use with ROM version.

Shorthand could be adapted to run with the ROM version of Applesoft II. The addresses in Shorthand would have to be changed. I do not have access to a ROM card and so do not know the addresses. But if the ROM version is just a relocated RAM version, the addresses in Shorthand and table just need \$C800 added to them.

Shorthand does not use all of the control keys because some have special functions. These functions are shown in Table 1. If you do not mind losing these

functions, these keys can be used also. The choices for which command is tied to which key is shown in the program listing. If you do not like my choices, you can change the command addresses stored in Table 2. The addresses are for the RAM version and will not work for the ROM version.

Use Of Shorthand

Shorthand is relocatable and can be placed anywhere in memory. I normally load it at \$300—\$3AE, which is where I assembled it. But it can be placed anywhere. Applesoft's HIMEM: can be used to protect some upper memory.

Example:

A 32K system without DOS can have Shorthand loaded at \$7F51-7FFF and then HIMEM: can be set to 32593. So to bring up Shorthand use the following steps:

1. LOAD and RUN the Applesoft TAPE
2. Enter the monitor by pressing RESET or do a CALL—151
3. Type
300.3AER
or type
7F51.7FFFR
4. Start tape with Shorthand on it and press RETURN, stop the tape when it has loaded
5. Type
OG
Press Return
6. Type
POKE 1144,0
Press RETURN

7. If Shorthand is at \$300—\$3AE type
POKE 56,0; POKE 57,3
If Shorthand is at \$7F51—\$7FFF type
POKE 56,81; POKE 57,127
8. Press RETURN
9. If Shorthand is at 7F51 type
HIMEM: 32593
Press RETURN

Another good place to store Shorthand is between Applesoft II and your program. The problem is that Applesoft's LOMEM: does not set the lowest memory used by Applesoft, but sets the point at which Applesoft will start storing variables. But the monitor can be used to set pointers. To do this the following steps are used:

1. LOAD and RUN the Applesoft II tape
2. Enter the monitor by pressing RESET or do a CALL—151
3. Type
3000.30AER
4. Start the tape with Shorthand on it and press RETURN
When it has loaded stop the tape.
5. Type
67:B0 30
Press RETURN
6. Type 30AF:0
30AF:0
Press RETURN
7. Type
OG

```

1000 *****
1010 * APPLESOFT II SHORTHAND *
1020 *
1030 *
1040 * BY ALLEN J LACY *
1050 * SEPT 1979 *
1060 *
1070 *****
1080 *
1090 ZP .EQ $1E R15 OF SWEET16
1100 *
1110 *****
1120 *
1130 * LOCATIONS 478-47F NOT USED BY *
1140 * SCREEN DISPLAY SEE MICRO # 8 *
1150 *
1160 *****
1170 *
1180 SW .EQ $0478 SWITCH
1190 CT .EQ $0479 CHAR COUNT
1200 XSAV .EQ $047A
1210 YSAV .EQ $047B
1220 POIN .EQ $047C POINTER
1230 ZPS .EQ $047E
1240 *
1250 RKEY .EQ $FD1B KEY READ CODE
1260 SW16 .EQ $F689 SWEET16
1270 .OR $300
1280 *
1290 *****
1300 *
1310 * START LOCATION OF SHORTHAND *
1320 *
1330 *****
1340 *
1350 SH STX XSAV SAVE X REG
1360 STY YSAV SAVE Y REG
1370 PHA PHA
1380 LDA ZP LDA ZP
1390 STA ZPS STA ZPS
1400 LDA ZP+1 LDA ZP+1
1410 STA ZPS+1 STA ZPS+1
1420 *
1430 *****
1440 *
1450 * SWEET16 IS USED TO STORE KP *
1460 * PROGRAM COUNTER IN $1E $1F *
1470 * AND THIS IS USED TO FIND *
1480 * THE LOCATION OF THE TABLE IN *
1490 * SHORTHAND *
1500 *
1510 *****

300- 8E 7A 04
303- 8C 7B 04
306- 48
307- A5 1E
309- 8D 7E 04
30C- A5 1F
30E- 8D 7F 04

*****
311- 20 89 F6
314- 00
315- AD 78 04
318- D0 3B
31A- 68
31B- 20 1B FD
31E- 48
31F- C9 9B
321- 90 14

*****
323- 68
324- AE 7A 04
327- AC 7B 04
32A- 48
32B- AD 7E 04
32E- 85 1E
330- AD 7F 04
333- 85 1F
335- 68
336- 60
337- 29 7F
339- 0A
33A- 69 64
33C- A8
33D- C8
33E- B1 1E

*****
340- F0 E1
342- 8D 7D 04
345- 88
346- B1 1E
348- 8D 7C 04
34B- A9 FF

*****

1520 *
1530 JSR SW16 ENTER SWEET16
1540 .HS 00 LEAVE SWEET16
1550 KP .EQ * KNOWN POINT
1560 LDA SW CHECK SW
1570 BNE NBYT RESTORE ACC
1580 PLA RESTORE A KEY
1590 JSR RKEY STORE KEY VAL
1600 PHA STORE KEY VAL
1610 CMP #9B CONTROL KEY?
1620 BCC CTR
1630 *
1640 *****
1650 *
1660 * IF NOT A CONTROL JUST RETURN *
1670 *
1680 *****
1690 *
1700 RET RESTORE KEY
1710 RT RESTORE X REG
1720 LDY YSAV RESTORE Y REG
1730 PHA
1740 LDA ZPS RESTORE ZERO
1750 STA ZP PAGE LOCATIONS
1760 LDA ZPS+1
1770 STA ZP+1
1780 PLA
1790 RTS
1800 CTR AND #7F WHICH KEY
1810 ASL TIMES 2
1820 ADC #TAB-KP OFFSET FROM KP
1830 TAY
1840 INY
1850 LDA (ZP),Y LOAD ENTRY
1860 *
1870 *****
1880 *
1890 * IF VALUE OF THE HIGH BYTE IN *
1900 * TABLE IS 0 THEN RETURN THE *
1910 * CONTROL CHAR ELSE SET UP TO *
1920 * RETURN THE CHARACTERS FROM *
1930 * APPLESOFT'S INTERNAL TABLE *
1940 *
1950 *****
1960 *
1970 BEQ RET IF 0 RETURN
1980 STA POIN+1 STORE IN POIN
1990 DEY
2000 LDA (ZP),Y
2010 STA POIN
2020 LDA #SFF SET SW

```



```

34D- 8D 78 04      STA SW      2030
350- A9 00 00      LDA #0      2040
352- 8D 79 04      STA CT      2050
2060 *
2070 *****
2080 *
2090 * NBYT IS USED TO PASS THE
2100 * CHARACTERS FROM THE TABLE IN
2110 * APPLESOFT AS IF THEY WERE
2120 * TYPED IN
2130 *
2140 *****
2150 *
2160 NBYT PLA
2170 LDY CT      LOAD CHAR CT
2180 LDA POIN    STORE POIN IN
2190 STA ZP      ZERO PAGE
2200 LDA POIN+1
2210 STA ZP+1
2220 LDA (ZP),Y  LOAD NEXT CHAR
2230 CMP #80     LAST CHAR?
2240 BCS END
2250 ORA #80
2260 INC CT
2270 BNE RT
2280 END
2290 PHA
2300 LDA #0
2310 STA SW
2320 PLA
2330 BNE RT
2340 *****
2350 *
2360 * TABLE TO STORE ADDRESSES OF
2370 * COMMANDS IN APPLESOFT II
2380 *
2390 * WILL HAVE TO BE CHANGED FOR
2400 * ROM VERSION
2410 *
2420 *****
2430 TAB
2440 .DA $8F9 @
2450 .DA $A3B A CALL
2460 .DA $000 B PEEK
2470 .DA $000 C
2480 .DA $8EF D
2490 .DA $8D3 E TEXT
2500 .DA $000 F FOR
          G

```

Symbol Table

ZP	001E	SW	0478	CT	0479
XSAVE	047A	YSAV	047B	POIN	047C
ZPS	047E	RKEY	FD1B	SW16	F689
SH	0300	KP	0315	RET	0323
RT	0324	CTR	0337	NBYT	0355
END	0370	TAB	0379	LS	03AF

Table 1

Control U	-->
Control H	<--
Control M	RETURN
Control J	Line feed
Control G	BELL
Control X	Kill input line
Control C	Stops a running program
Control D	Is used by DOS

Press RETURN

8. Type
NEW
Press RETURN
9. Type
POKE 1144,0
Press RETURN
10. Type
POKE 56,0:POKE 57,48
Press RETURN

Shorthand will now be tied in.

Step 5 sets the pointer which tells Applesoft II where to start storing a program to \$3080. Step 6 sets the byte just below the start point to 0, I do not know why Applesoft wants this, but it will bomb if it is not done. Step 8 causes Applesoft to reset the rest of its pointers to reflect the new start point.

Now every time you want to type one of the commands stored in the table just press the control key and another key at the same time.

Example:

To enter INPUT press the control key at the same time as the I.

I have made labels for my keyboard showing which command is under which key. To return full control to the keyboard, use the command IN 0. To turn Shorthand back on just POKE the correct values back into 56 and 57. Shorthand does not have to be turned off when you are finished programming and want to run a program, unless the program wants for

input one of the control keys which Shorthand uses. I normally set the hooks when I bring up Applesoft and leave them set.

The routine should work with DOS. I do not have DOS so these techniques are not tested. Since DOS communicates with the rest of the system via the input and output hooks at \$36—39, you can not set the hooks to tie in shorthand without turning off DOS. But DOS has its own internal hooks. Unfortunately the hooks are at different places for different memory sizes. In a 48K system the input hook is at \$A998, \$A999 (22120, 22119 decimal). For smaller systems subtract 48K—X from the numbers, where x is the memory size. The above information came from Exploring the APPLE II DOS by Andy Hertzfeld in MICRO 9. So POKE the address of Shorthand in the DOS hooks

Another way that should work is to turn DOS off by the use of the following steps.

1. After bringing up Applesoft and loading Shorthand type
PR O:IN 0
Press RETURN
2. Use POKes to set 56 and 57 if Shorthand is at \$300
POKE 56,0:POKE 57,3
3. When you are finished type
CALL 976
Press RETURN

Step 1 turns DOS off. Step 2 turns shorthand back on. DOS will not be on at the same time as Shorthand.

Table 2

8D0 END	8D3 FOR	8D6 NEXT	8DA DATA
8DE INPUT	8E3 DEL	8E6 DIM	8E9 READ
8ED GR	8EF TEXT	901 HLIN	905 VLIN
909 HGR2	90D HGR	910 HCOLOR=	917 HPLOT
91C DRAW	920 XDRAW	925 HTAB	929 HOME
92D ROT=	931 SCALE=	937 SHLOAD	93D TRACE
942 NOTRACE	949 NORMAL	94F INVERSE	956 FLASH
95B COLOR=	961 POP	964 VTAB	968 HIMEM:
96E LOMEM:	974 ONERR	979 RESUME	97F RECALL
985 STORE	98A SREED=	990 LET	993 GOTO
997 RUN	99A IF	99C RESTORE	9A3 &
9A4 GOSUB	9A9 RETURN	9AF REM	9B2 STOP
9B6 IN	9B8 WAIT	9BC LOAD	9D0 CONT
9D4 LIST	9D8 CLEAR	9DD GET	9E0 NEW
9E3 TAB (9E7 TO	9E9 FN	9EB SPC (
9EF THEN	9F3 AT	9F5 NOT	9F8 STER
9FC +	9FD -	9FE *	9FF /
A00 +	A01 AND	A04 OR	A06 >
A07 =	A08 >	A09 SGN	A0C INT
A0F ABS	A12 USP	A15 FRE	A18 SCRNI
A1D RDL	A20 POS	A23 SQR	A26 RND
A29 LOG	A2C EXR	A2F COS	A32 SIN
A35 TAN	A38 ATN	A3B PEEK	A3E LEN
A42 STR\$	A46 VAL	A49 ASC	A4C CHR\$
A50 LEFT\$	A55 RIGHT\$	A5B MID\$	

Classified Ads

TEXTPRO: Hardware, Firmware and powerful Software. Apple II plug-in addition for (1) Text editing, (2) Screen editing, (3) programming Upper/Lower case, Powerful Super-mon-3 2716 EPROM Monitor I.C. Complete: 45 page documentation, text-processing software (disk) and all hardware. 15-minute installation. Call collect for information. \$349.

TextPro Corporation
1367 Post St. Studio 19
San Francisco, Ca 94109
(415) 775-5708

PET MACHINE LANG. GUIDE: Comprehensive manual to aid the machine language programmer. More than 30 routines are fully detailed so that the reader can put them to immediate use. For either Old or New ROMs. \$6.95 plus .75 postage. VISA or Mastercharge accepted. Order from:

Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510

APPLE Jewelry: Tie Tack, Stick Pin, or Lapel Pin. 1" diameter. Your choice \$4.95 plus 45¢ postage. Quantity discounts available for 10 or more pieces. Order from:

C. Armstrong
P.O. Box 15614
Columbus, Ohio 43215

The Life Dynamic Transformation Experience on the Apple II with 48K. In Applesoft and Machine Language; Apple II Plus, Disk II. Unique! This program is designed for all those people who desire to experience self-transformation, life-awareness, making relationships work, and "getting your act together", but do NOT desire to pay est or Lifespring or any of the other "trips" of the Human Potential Movement, the \$300 or so. Includes game-playing as a means of a fun way to increase awareness. \$15.95 (disc with instructions) Order from:

Avant-Garde Creations
P.O. Box 30161 Dept. MC
Eugene, OR. 97403

The Value of 16 Bits

Several years ago, the guest speaker at the local computer club, a gentleman from Texas Instruments, talked about the importance of the size of a microprocessor. Using all kinds of charts, tables, and various rather logical sounding arguments, he determined that 8 bit micros did not make any sense and would never find much popularity or application! A 4 bit micro is all that is required in most process control situations, and anyone wanting to do real computer type stuff—number crunching, assembling, text processing—would much prefer a 16 bit micro. Conclusion: the 8 bit micro was doomed. Well, hundreds of thousands of 8 bit microcomputers later, it is obvious that there is a market for the 8 bit micro. Isn't 20/20 hindsight wonderful!

Actually, I did not buy this thesis at the time it was presented. I had worked on a number of projects with either minis or a precursor of the micros, and had discovered a number of instances in which an 8 bit processor was superior to its bigger brother. Does this seem strange? Let's examine the details.

One obvious type of application, in which we all participate to some degree, is any form of word processing. How many bits does it normally take to represent the normal alphanumeric and special symbols that we use in everyday writing, BASIC, assembler programming, and so forth? ASCII defines 128 characters, including a bunch of specialized control codes, and that seems to be enough for most applications. Even if you want to add special sets, such as greek for APL, the total number of unique codes required is normally going to be less than 256 decimal. Can you imagine

a keyboard to generate more than 256 characters? Since 8 bits can be used to represent 256 unique values, it is adequate for this work. In fact, it is ideal. A 16 bit machine either must ignore half of each byte, which is of course wasteful and essentially reduces it to an 8 bit machine, or must pack two 8 bit bytes into each 16 bit word. And then it must, of course, unpack the two bytes for processing, repack them again, and so forth. Therefore, the 8 bit micro is perfect for most word processing based applications. Since this single application category must account for a large percentage of the systems being purchased today, the strength of the 8 bit micro should not be surprising.

Another application I worked on used a high speed photo scanner to digitize material for use in newspaper production—halftones and text. The scanner produced 8 bit chunks of data. The microcomputer was 16 bit based, and a lot of overhead was spent in packing and unpacking data, making records come out to an integral number of words, and other such nonsense. While the fact that 8 bits were appropriate to this particular application may have been pure serendipity, I am sure that there are numerous process control types of application which have a similar data range and which could best be served by the 8 bit micro.

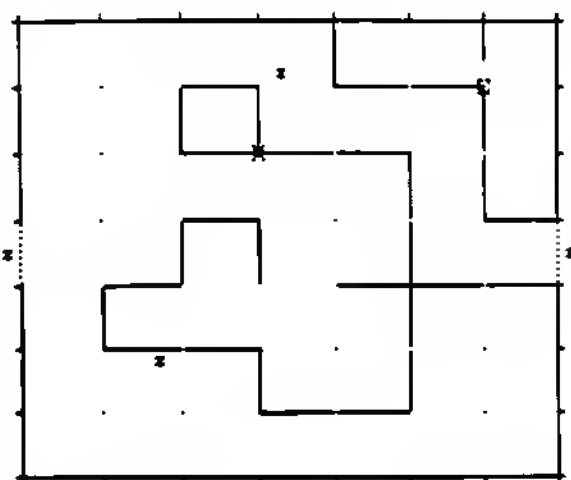
Okay, how about number processing. Surely the 16 bit micro is better at performing math functions than the 8 bit micro. True, there is some advantage to a 16 bit micro if your application requires a lot of number crunching. 16 bit math operations can handle twice as much data as 8 bit ones. But, the savings may be minimal. In many numeric calculations, the

amount of code and time spent actually performing math functions may be insignificant relative to the amounts required to do all of the other programming steps required—the set up, testing one bit, branching, subroutine jumps, and so forth. So, while there will probably be a time improvement with a 16 bit micro in heavy math programs, the savings may not be as great as initially imagined.

Where does the 16 bit computer excel then? I am not sure that, in general, it does. Given the generally higher cost of the micro, the higher cost and complexity of a 16 bit data bus, and so forth, the 16 bit must justify itself for a particular application. It is not a generally "better" solution. There are some features of a typical 16 bit micro that would be nice to have in the 8 bit as well. This is particularly true in improved addressing capabilities. Since the address space of most 8 bit micros is actually 16 bits, it would make sense in many instances to be able to handle the full range of address space with 16 bit registers. In the 6502, a number of 16 bit addressing modes are already supported. The two main places where the 8 bit limit is restrictive are in the relative branches and in the indexed instructions. The "proposed" 6516 discussed by Randall Hyde in this issue shows how the benefits of a 16 bit micro can be combined with the strengths of the 8 bit micro to form a superior computer. It is interesting to note, however, that many of the improvements are **not** based on 16 bits, but are independent enhancements. My latest intelligence suggests that the initial statement in the referenced article—"Synertek is almost ready to ship the SY6516"—is a bit optimistic. But, if we all call and ask our Synertek Reps about this superior product, maybe we can get some action!

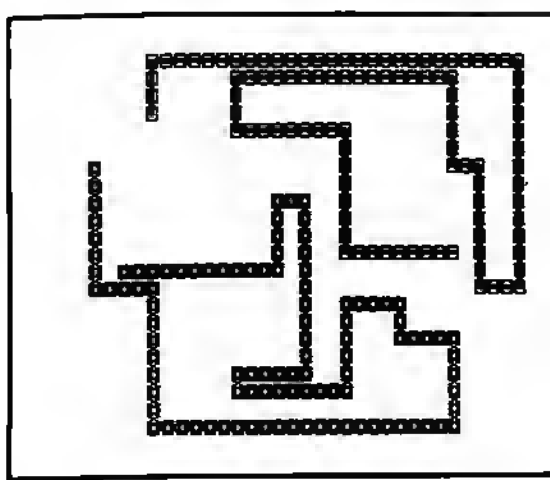
Robert M. Gandy

Software for the Apple II



SCORE: 108

DYNAMAZE—a dazzling new real-time game. You move in a rectangular game grid, drawing or erasing walls to reflect balls into your goal (or to deflect them from your opponent's goal). Every ball in your goal is worth 100 points, but you lose a point for each unit of elapsed time and another point for each time unit you are moving. Control the speed with a game paddle: play as fast as ice hockey or as slowly and carefully as chess. Back up and re-play any time you want to; it's a reversible game. By Don Stone. Integer Basic (plus machine language); 32 K; \$9.95.



SCORE: 105

ULTRA BLOCKADE—the standard against which other versions have to be compared. Enjoy Blockade's superb combination of fast action (don't be the one who crashes) and strategy (the key is accessible open space—maximize yours while minimizing your opponent's). Play against another person or the computer. New high resolution graphics lets you see how you filled in an area—or use reversibility to review a game in slow motion (or at top speed, if that's your style). This is a game that you won't soon get bored with! By Don Stone. Integer Basic (plus machine language); 32 K; \$9.95.

What is a **REVERSIBLE GAME**? You can stop the play at any point, back up and then do an "instant replay", analyzing your strategy. Or back up and resume the game at an earlier point, trying out a different strategy. Reversibility makes learning a challenging new game more fun. And helps you become a skilled player sooner.

WORLD OF ODYSSEY—a new adventure game utilizing the full power of Disk II, which enables the player to explore 353 rooms on 6 different levels full of dragons, dwarfs, orcs, goblins, gold and jewels. Applesoft II 48K; \$19.95 includes diskette.

PEROUACKEY—an exciting vocabulary game which pits the player against the clock. The object of the game is to form words from a group of 10 letters which the computer chooses at random. The words must be 3 to 10 characters in length with no more than 5 words of any particular length. Each player has only 3 minutes per turn. The larger the words the higher the score. Applesoft II 16K; \$9.95.

APPLESHIP—is a naval game in which two players enter their ships in respective oceans. Players take turns trying to blast their opponent's ships out of the water. The first player to destroy their opponent's ships may win the game. A great low-res graphics game. Applesoft II 32K; \$14.95.

Available at your
local computer store

Call or write for our free
SOFTWARE CATALOG

Apple II is a registered
trademark of
Apple Computer, Inc.

DEALER INQUIRIES INVITED

POWERSOFT, INC.

P. O. BOX 157
PITMAN, NEW JERSEY 08071
(609) 589-5500

Programs Available on Diskette
at \$5.00 Additional

- Check or Money Order
- Include \$1.00 for shipping and handling
- C.O.D. (\$1.00 add'l. charge)
- Master Charge and VISA orders accepted
- New Jersey residents add 5% sales tax

The APPLE Stripper

One of the classic dilemmas in BASIC has to do with REMarks. If you use them, they take up space and time. If you do not use them, the code is hard to understand. This program resolves the problem. It permits you to generously REMark your program for documentation purposes and then remove the REMarks for the run-time version.

Bill Crouch
P.O. Box 926
Long Beach, CA 90801

As a writer of custom business software for the APPLE computer, I kept running into the same conflict; good programming style insisted that I document my programs with frequent REMark statements. My customers would have a hard time understanding or changing my programs if I did not.

On the other hand, large business programs use a great deal of memory and every byte is precious. The Applesoft manual tells us that the statement: 130 THIS IS A COMMENT uses up 24 bytes of memory. In a large program, a lot of memory will be taken by REMs, leaving less for arrays and program operation. It also means more frequent waits while the machine "housecleans" its string space.

The answer is obvious; write the program with REMarks and then remove them in the final working version. If changes are needed, make them on the version with REMarks and then remove the REMs again after the bugs have been corrected.

Removing REMs by hand took too long so I wrote a simple program to do it for me. It is disk based and will work on any APPLE with a disk drive.

Program Requirements

To use this program you need only observe a couple of simple rules. First, NEVER GOTO or GOSUB to a REMark. Always GOTO or GOSUB to the first line of code after the REMark.

Secondly, for maximum benefits, put your REMarks on a separate line rather than at the end of a line of code. This program only eliminates those lines where a REM is the first thing in the line.

```
10 REM
    REM KILLER

20 REM BY BILL CROUCH
30 REM PO BOX 926
40 REM LONG BEACH CA 90801

50 PRINT CHR$(4);"MON I,O,C"
60 DIM ARRAY(1000)
70 ONERR GOTO 240
80 X = 0
90 REM

    READ TEXT FILE

100 HOME : REM CLEAR SCREEN
110 PRINT CHR$(4);"OPEN PROG.FILE"
120 PRINT CHR$(4);"READ PROG.FILE"
130 INPUT L$: REM GET A LINE FROM DISK
140 IF LEFT$(L$,5) = "63000" GOTO 250: REM CHECK FOR END OF TEXT
150 IF L$ = "" GOTO 130: REM ELIMINATE NULL STRINGS
160 LN = VAL (L$):LN = INT (LN): REM SAVE LINE NUMBER
170 IF LEFT$(L$,1) = "" THEN L$ = RIGHT$(L$, (LEN (L$) - 1)): GOTO 1
    70
180 IF LEN (L$) < 2 GOTO 130: REM IF LINE USED UP GET ANOTHER
190 IF ASC (L$) < 65 THEN L$ = RIGHT$(L$, (LEN (L$) - 1)): GOTO 170
200 IF LEFT$(L$,3) = "REM" THEN X = X + 1:ARRAY(X) = LN: REM KEEP TRAC
    K OF REMS
210 IF X > 995 GOTO 250: REM STAY WITHIN ARRAY
220 GOTO 130: REM DO IT ALL AGAIN
230 REM

    WRITE STRIP FILE

240 IF PEEK (222) < > 5 GOTO 130: REM CHECK FOR OUT OF DATA ERROR
250 PRINT CHR$(4);"CLOSE"
260 POKE 216,0: REM CLEAR ONERR GOTO FLAG
270 IF X = 0 GOTO 340: REM NO REMS IN PROGRAM
280 PRINT CHR$(4);"OPEN STRIP.FILE"
290 PRINT CHR$(4);"WRITE STRIP.FILE"
300 FOR Y = 1 TO X
310 PRINT ARRAY(Y): REM SAVE LINE # OF REM
320 NEXT Y
330 PRINT CHR$(4);"CLOSE"
340 END

]
]PR#0
```


How to Use the Programs

There are two separate programs. The first, XFILE.MAKER, must be appended to the end of your program. You could type it in yourself or, better still, use the merge routine on the DOS 3.2 Master. The only requirement is that line 63000 be after the last line of your program. It tells the next program that it is done.

You start the process with the command "RUN 63000"

You should have both programs on their own diskette with plenty of space for their text files. If REM KILLER is not on the same diskette with XFILE.MAKER, remove line #63130.

XFILE.MAKER will convert your program into a text file and then run REM KILLER. REM KILLER then reads the text file, makes a list of REMs and then writes them off as STRIP.FILE.

By the way, certain characters in your program will cause the computer to say EXTRA IGNORED during the running of REM KILLER. You can ignore it too.

When it is done, load your original program and EXEC STRIP.FILE. Every line which is a REMark will be removed. Then save the stripped program.

Of course also save a copy of your original program. The first program I used this on was part of a trucking company

package. It saved me over 2400 bytes.

How it Works

XFILE.MAKER clears the screen with line 63050 and squashes the listing to suppress extra carriage returns with line 63060.

The rest of the program writes your program to the disk as a text file. Line 63130 calls REM killer.
(Note: CHR\$(4) is the same as CTRL D and is required before every APPLE disk command.)

REM KILLER: Line 60 sets up an array in which REMs are saved. It now allows for 1000 REMs which probably is too many. If you have memory limitations, you may reduce this number and the corresponding one on line 210.

Line 140 checks for the end of your file and is the reason line 63000 is required in XFILE.MAKER.

Lines 150-190 get rid of null lines and all non-alpha characters. Line 200 then sees if the first alpha string is REM. If so, it saves the number in the array.

Lines 240-340 save the approximate line numbers as a text file called STRIP.FILE.

When you EXEC STRIP.FILE, the line numbers are printed just as it you had typed them yourself. And the REMark lines are eliminated.

XFILE.MAKER

```
63010 REM
      BY BILL CROUCH

63020 REM PO BOX 926
63030 REM LONG BEACH CA 90801

63040 REM APPEND TO END OF PROGRAM
63050 CALL - 936
63060 POKE 33,33: REM FORMAT LISTING
63070 PRINT CHR$(4);"MON I,O,C": REM LET US SEE IT WORK
63080 PRINT CHR$(4);"OPEN PROG.FILE"
63090 PRINT CHR$(4);"WRITE PROG.FILE"
63100 LIST 0,63000
63110 PRINT CHR$(4);"CLOSE"
63120 TEXT
63130 PRINT CHR$(4);"RUN REM KILLER"
63140 END
63150 REM
```

CHANGE CHR\$(4) TO CTRL D FOR INTEGER PROGRAMS

Classified Ads

SYM/KIM Appendix \$4.25 postpaid, (see MICRO, 19:68 for description). First bk. of KIM: \$10. Combo Appendix & 1st bk: \$13.50. SYM-1 Hardware Theory Manual supplements, SYM-1 Ref. Manual \$6.00. Order from:

Robert A. Peck
P.O. Box 2231
Sunnyvale, CA 94087

Color Graphics for OSI C1P or Superboard II. Details on modifications enable operation in several color graphic modes. Easy installation. Includes colored-video game for demo, and checkout that runs on 4K RAM systems. Low-cost hardware not included. Send \$8.65 to:

Tom Hofman
873 Dorset Drive
Knoxville, TN 37919

100 percent PET disk oriented Macro Assembler/Text Editor (MAE) Development software. Includes a new version of our ASSM/TED written specifically for the 32K new ROM PET and 2040 disk drive. Features macros, conditional and interactive assembly, and includes a relocating loader program. \$169.95 includes diskette and manual. Send \$1.00 for details. Eastern House Software
3239 Linda Drive
Winston-Salem, NC 27106

ATTENTION SYM-1 OWNERS: Modify your SYM to have 8K of 2114 RAM on board with the W7AAY piggyback RAM board. Bare board plus instructions is \$5.00 (plus a self addressed, stamped envelope per board. The new W7AAY RAE 1/2 board plugs into and extends ROM socket p3 so that the two chip version of RAE only needs one SYM socket. Completely assembled, with instructions for \$10.00 postpaid in the USA. Order from:

John M. Bialock
P.O. Box 39356
Phoenix, AZ 85069

Advertise in Micro

An ad in this section may be run for \$10. per month. Should not exceed 6 lines. One ad per person (or co.) per month. Must relate to 6502 industry, be prepaid. Deadline: 25th of month.

nibble

TWO DOLLARS
MAY 1960

MODEL 10.
A SOFTWARE GUIDE for high quality Applications
Programs for your Home and Business.

MIDDLE 10. *A REFERENCE GUIDE* to new Programming Methods.

TABLE 10.
A BUYERS GUIDE for making purchase decisions on new products.

A CONSTRUCTION PROJECT COOKBOOK for adding function and value to the system you already own.

MODEL 16:
A COMMUNICATIONS CLEARING HOUSE for users,
vendors, and associations.

Examples of upcoming articles:

- ☐ Modeling and Forecasting Your Business ☐ Build a Two-Tape Controller for \$12
☐ Arcade Shooting Gallery — Save Your Quarters! ☐ Data Base Management
System I, II, III

Try a NIBBLE

nibble

BOX 325 Lincoln, Mass. 01773

I'll try NIBBLE!

Enclosed is my \$15 for 8 issues.

☐ check ☐ money order

Name _____

Address _____

City _____

State _____ Zip _____



"As manager of Personal Computers, here at Computerland of San Francisco, evaluating new software products is part of my job. With all the word processors on the market today, I choose EasyWriter for my business and personal use."

—Karen Dexter Weiss

EasyWriterTM

80 COLUMNS OF WORD PROCESSING POWER FOR YOUR APPLE II COMPUTER

Finally . . . INFORMATION UNLIMITED SOFTWARE is able to bring you a complete word processing system for the Apple II. The new EasyWriter system gives you **80 columns of upper and lower case characters** for your Apple's video display, using the new SUP'R'TERM 1 board!



You can purchase the new EasyWriter 80 column word processing system as a complete hardware and software package directly from our new office in California, or from your local computer dealer.

A long time ago, we decided to bring you the best simple-to-use and understand tools for your computer system. Today we've taken another stride in that same direction. It took some doing, in both hardware and software, but we think you'll agree that for the buck, no one can touch us.

Check it out:

- 80 Columns on the Screen!
- Upper & Lower Case!
- Global Search & Replace!
- Underlining!
- Bidirectional Printing!
- Incremental Spacing!
- File Appending!
- 50 Pages of Text Per Disk!



Information Unlimited Software, Inc.
793 Vincente St.
Berkeley, CA 94707
(415) 525-4046

- EasyWriter is a TM of Cap'n Software, Inc.
- Apple II is a TM of Apple Computers, Inc.

See The System at the 5th West Coast Computer Faire.

Graphics and the Challenger C1P, Part 4

This continuing series on Graphics and the Challenger shows how to apply the material to create pictures and demonstrates how this may be used in Computer Aided Instruction.

William L. Taylor
246 Flora Road
Leavittsburg, OH 44430

Computers are well suited for use in an educational environment, whether this is in a class room at a local high school, college, or in an Industrial training seminar. The computer can aid the instructor or can be used as an individual instructor. With the introduction of the micro processor and the number of low cost personal computers that are owned and used by individuals as a hobby, the computer must be considered as a training tool for use in the home environment.

Children seem fascinated by computers and are equally fascinated by any device that has a keyboard. If the computer has any form of graphics display, either animated or still, they seem even more delighted to experiment with the device. This leads to the point that if children are drawn to the computer, then the computer, if programmed to be a teaching aid, can be a valuable tool in their education.

With this evidence I decided to try to develop a program that combines the elements that have the most attraction for children. Also, through this method, the program will at the same time be an educational tool.

The program, which I will call "Picture" was developed to be a teaching aid in the development and spelling of English words. The program uses Graphics to draw a picture of several objects. Then the child is asked to spell the different parts of the picture that have been displayed. The child tries to spell the names of the objects displayed, and the computer displays the answer "Right" or "Wrong" on the screen in large letters.

In Part 3 of this series ("Graphics and the Challenger C1P"), we described the

features of the C1P. We developed some programs using Basic and Machine Language, in combination, to further explore the Graphics capabilities of the C1P. Many techniques were discussed and many Basic functions and statements were used in our example programs. This time let's continue with our graphics development and try a new programming approach.

This article has a two-fold purpose. First to continue our discussion of how to use the Graphics of the OSI Challenger C1P, and to secondly present a working program using the Graphics techniques

in a Computer Assisted Instruction program (CAI). The program in this part will be used as a CAI tool and will be treated as an example program. *This program, by no means, is complete.* That is, it can be expanded by the user. The program simply is a pure example of how to develop graphic plots: get these characters out to the monitor screen. Combining these Graphics with a program is a useful tool in the hands of the enterprising programmer. From the techniques that are presented in this example, the user will more fully understand how to develop such programs of his own.



Program Description

Let's start with a description of the "picture" program. First, what the program does is to generate a picture on the monitor screen. This picture is shown in the video memory map plotting chart in Figure 1. Notice that we have developed routines in the program that will POKE characters from the Graphics Set in the Character Generator ROM out to the monitor screen at the locations shown on the chart. In part 3 of this series, I gave a similar video memory chart. This time we will use the chart as in Figure 1. Notice that in the chart, we have drawn the picture that we wish to POKE out to the screen. All the memory locations can now easily be found, and routines written to accomplish the end task. Such a routine is located in the program between lines 10000 and 10420. This routine is used to draw the House, the Airplane, the Sun, the Man and the Car in the picture. All the parts of the picture were built from the Graphics elements in the Character Generator ROM. A list of these character elements appears in the upper left corner of Figure 1.

Please examine the program listing, starting at line 10000. Take the value in the statement line, For A = 53606 To 53926 Step 32. From these statement values find the corresponding value on the video memory map chart in Figure 1. It will be found that when the statement line at 10000 is compared to the memory map, you will be able to see just what the For-Next loop does. The Characters will be poked to these locations. Examine the program completely from line 10000 to 10420 to see how each unit works compared to the map. This example should give you a clear understanding of how to use the memory chart so that you can develop routines for your own program.

The program "Picture" contains two other Graphics Routines. These routines are used with the program to inform the user (or student) if he has identified and correctly spelled and element on the picture, or if he has identified and incorrectly spelled the object. These routines display the words: Right and Wrong, respectively. These words are in large graphic format at the top of the C1P's monitor screen. A video memory location map of these elements are in Figures 2 and 3. Please review these two figures for the memory locations. The subroutines for these graphic displays are located beginning at line 20000 for the word "Right" and at line 5000 for the word "Wrong".

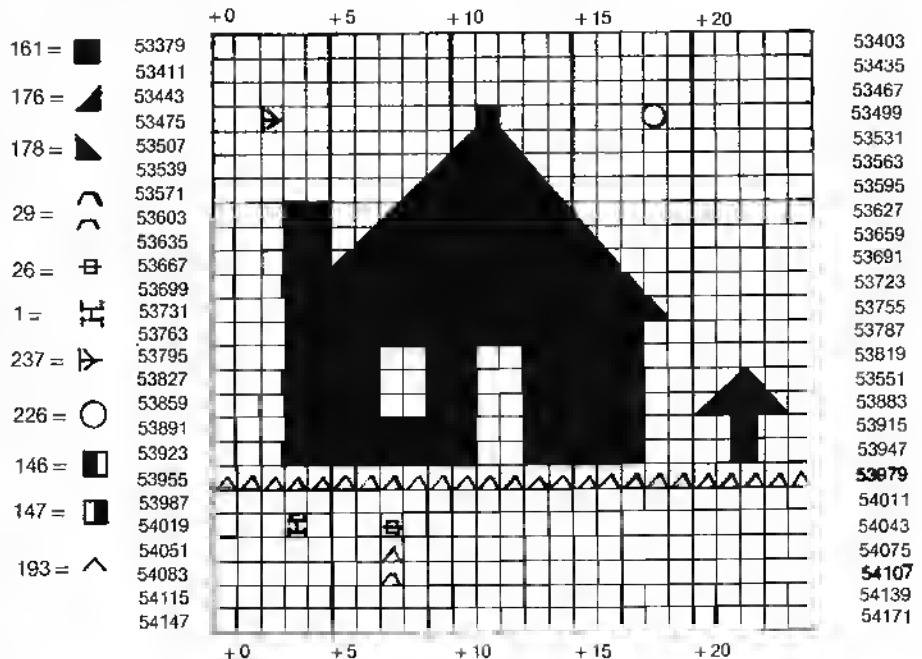
These two subroutines were developed in the same manner as the one for the picture. That is, the video memory locations were plotted on the video memory plotting chart. Next, the graphics elements that were needed to generate the characters were selected from the list of graphics elements and finally, routines were written to do the task of POKEing the elements out to the screen. Analyze

OK
LIST

```

1 GOSUB 8000
35 PRINT"**** PICTURE ****":PRINT
50 PRINT
60 PRINT" HELLO IM A COMPUTER    MY NAME IS
   CHALLENGER"
70 PRINT:PRINT
80 PRINT" WHAT IS YOUR NAME?"
90 INPUT A$
100 PRINT:PRINT"HELLO "A$;"  GLAD TO MEET YOU"
110 PRINT:PRINT" THIS IS A SPELLING GAME";A$
120 PRINT:PRINT" I WILL SHOW YOU A"
130 PRINT" PICTURE.  YOU ARE TO"
140 PRINT"TELL ME THE PARTS"
150 PRINT" THAT YOU KNOW"
155 FOR Q=1 TO 10000:NEXT Q
160 POKE 11,232:POKE 12,15
170 X=USR(X)
180 GOSUB 10000
190 PRINT:PRINT" DID YOU SEE THE PICTURE?"
200 PRINT" TELL ME THE PARTS":PRINT
210 PRINT:PRINT" SPELL THE PARTS THAT MAKE
   THE PICTURE"
220 INPUT B$
230 IF B$="ROOF" THEN A=2
240 IF B$="CHIMNEY" THEN A=2
250 IF B$="WINDOW" THEN A=2
260 IF B$="DOOR" THEN A=2
270 IF B$="YARD" THEN A=2
280 IF B$="ROOF" THEN A=2
290 IF B$="CHIMNEY" THEN A=2

```



C1P Video Memory Map in Decimal


```

300 IF B$="SUN" THEN A=2
310 IF B$="PLANE" THEN A=2
330 IF A<>2 THEN GOSUB 5000
335 IF A=2 THEN GOSUB 20000
500 GOTO 200
5000 FOR A=53541 TO 53637 STEP 32
5010 POKE A,161:NEXT A
5020 FOR A=53544 TO 53640 STEP 32
5030 POKE A,161:NEXT A
5040 POKE 53638,175:POKE53639,
177:POKE 53606,176:POKE 53607,178
5050 FOR A=53546TO53642STEP32
5060 POKEA,161:NEXTA
5070 POKE 53547,151:POKE 53548,161:
POKE 53579,150
5080 POKE 53580,175:POKE 53611,177:
POKE 53612,178
5090 POKE 53644,161
5100 FOR A=53550 TO 53646 STEP 32
5110 POKE A,161:NEXT A
5120 FOR A=53552 TO 53648 STEP 32
5130 POKE A,161:NEXT A
5140 POKE 53551,161:POKE 53647,161
5150 FORA=53554 TO 53650 STEP32
5160 POKE A,161:NEXT A
5170 FOR A=53556 TO 53652 STEP 32
5180 POKE A,161:NEXT A
5190 POKE 53587,178:POKE 53610,177
5200 FOR A=53590 TO 53654 STEP 32
5210 POKE A,161:NEXT A
5220 FOR A=53592 TO 53720 STEP 32
5230 POKEA,161:NEXT A
5240 FOR A=53591 TO 53719 STEP 64
5250 POKE A,161:NEXT A
5260 FOR T=1 TO 500:NEXT T
5270 X=USR(X)
5280 RETURN
8000 FOR Q=4072 TO 4095
8010 READ F:POKE Q,F
8020 NEXT Q
8030 DATA 169,32,160,8,162,0,157,0
8040 DATA 208,232,208,250,238,240
8050 DATA 15,136,208,244,169,208
8060 DATA 141,240,15,96
8070 RETURN
10000 FOR A=53606 TO 53926 STEP 32
10010 POKE A,161:NEXT A
10020 FOR A=53607 TO 53927 STEP 32
10030 POKE A,161:NEXT A
10040 FOR A=53517 TO 53672 STEP 31
10050 POKE A,176:NEXT A —
10060 FOR A=53519 TO 53716 STEP 33
10070 POKE A,178:NEXT A
10080 FOR A=53673 TO 53683
10090 POKE A,161:NEXT A
10100 FOR A=53642 TO 53650
10110 POKE A,161:NEXT A
10120 FOR A=53611 TO 53617
10130 POKE A,161:NEXT A
10140 FOR A=53580 TO 53584
10150 POKE A,161:NEXT A
10160 FOR A=53549 TO 53551
10170 POKE A,161:NEXT A
10180 POKE 53518,161
10190 POKE 53486,171
10191 FOR A=53704 TO 53716
10192 POKE A,161:NEXT A
10193 FOR A=53736 TO 53748
10194 POKE A,161:NEXT A
10195 FOR A=53768 TO 53780
10196 POKE A,161:NEXT A
10200 FOR A=53804 TO 53805
10205 POKEA,161:NEXT A
10207 FOR A=53836 TO 53837
10240 POKE A,161:NEXT A
10250 FOR A=53896 TO 53901
10260 POKE A,161:NEXT A
10270 FOR A=53928 TO 53933
10280 POKE A,161:NEXT A
10290 FOR A=53937 TO 53940
10300 POKE A,161:NEXT A
10310 FOR A=53905 TO 53908
10320 POKE A,161:NEXT A
10330 FOR A=53873 TO 53876
10340 POKE A,161:NEXT A
10350 FOR A=53841 TO 53844
10360 POKE A,161:NEXT A
10370 FOR A=53809 TO 53812
10380 POKE A,161:NEXT A
10390 FOR A=53800 TO 53801
10400 POKE A,161:NEXT A
10402 POKE53477,237
10405 POKE 53493,226
10410 FOR A=53955 TO 53979
10420 POKE A,193:NEXT A
10430 FOR D=1 TO 5000:NEXT D
10440 X=USR(X)
10450 RETURN
20000 FOR A=53509 TO 53637
STEP 32
20010 POKE A,161:NEXT A
20020 FOR A=53511 TO 53575
STEP 32
20030 POKE A,161:NEXT A
20040 POKE 53607,178:POKE53639,
177:POKE 53574,161:POKE 53510,161

```

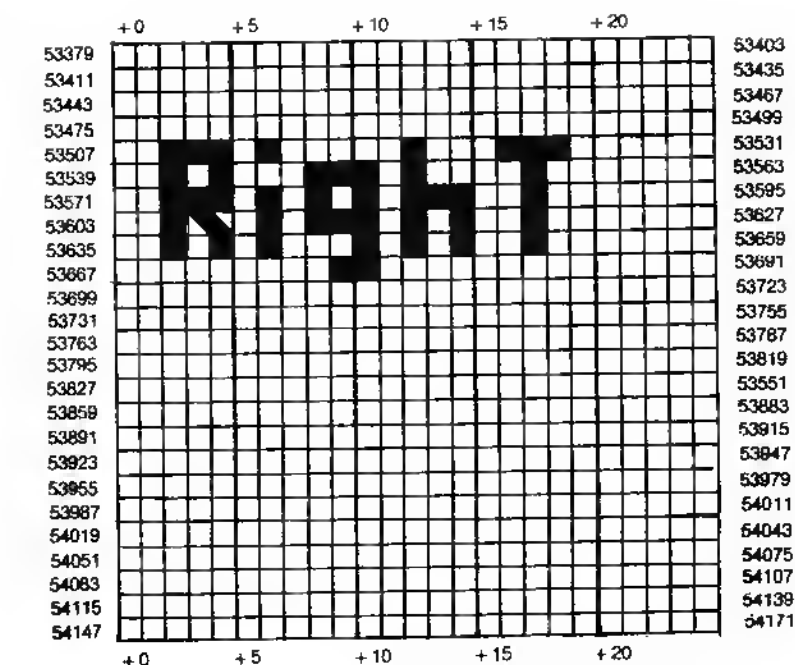
the video memory plotting charts of Figure 2 and Figure 3 along with the subroutines at lines 5000 and 20000 to see how the routines were plotted, written and used in the program.

The subroutine located between lines 8000 and 8070 is for loading the machine code routine into user memory. This routine is for the Fast Screen Erase routine used by the program to clear the screen whenever called. This subroutine will be called at line 1 in the Main Line Basic program. The routine for the fast screen erase has been included in the previous parts of this series. The reader should review these parts for a complete description of this routine.

Now that I have described the Graphics generating routines and how they were developed, let's continue with the Main-line BASIC program that uses the subroutines. The program from line 35 through 500 forms the BASIC CAI user program. This program is a demonstration of how to develop programs in which the user can be taught such things as spelling which is the purpose of this program, combined with the graphics presentation.

The For-Next loop at line 155 is used to give the user time to read the screen text just displayed. Statement line 160 sets the USR Vector to point to the Fast Screen Erase Machine Language routine located at OFE8 hex or 8168 decimal. Line 170 causes the program to jump to the machine language routine at OFE8 or 8168 decimal where the screen will be erased.

Input from the user is accepted at statement line 220. This data is stored in a

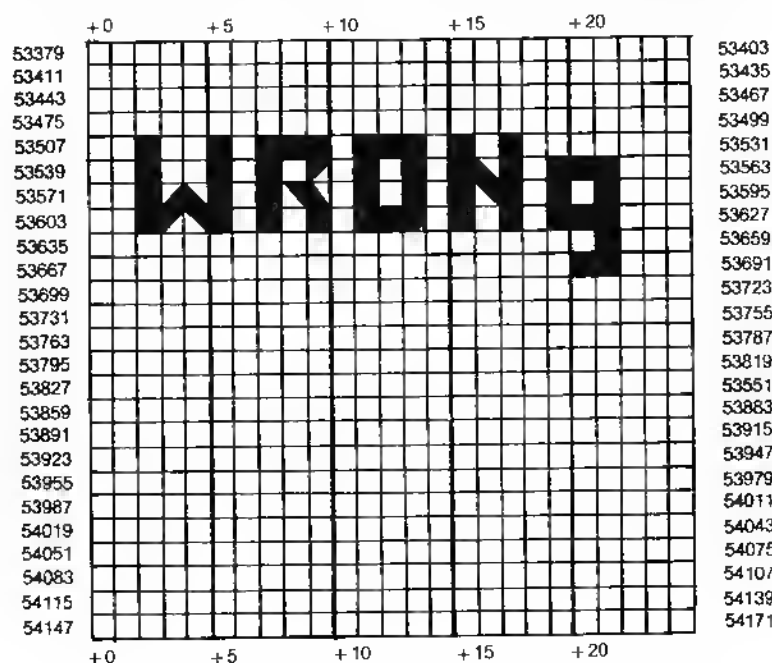


string variable, labeled B\$. The input string (B\$) is then compared in the string looking up table for a string match. If a match between the input and a table content is found, the information is then passed to the variable A as a decimal value. This value is then compared at Line 330 and Line 335 to check for a correct answer from the user. If a correct match was found in the string table, the A variable will force a GOSUB to Line 20000 where the answer word "Right" will be displayed for the user's answer. If a match did not occur in the string table, at

line 330 a GOSUB to line 5000 will cause the answer word "WRONG" to be displayed informing the user that the answer was not correct or was not an element in the picture. At line 500 a return to the beginning of the BASIC Main-Line program will cause a new pass through the program.

This program, as stated before, is not really complete, but an example to show how such a program can be constructed. This program can be expanded or modified by the reader. If you should desire to expand the picture display to include more objects, then these object names should be included in the string table.

The complete program including the Graphics routines and the fast screen erase routine is located at the top of the first 4K of user memory. If you have more memory and wish to expand the program, you will have to relocate this routine. Listing 2, shows the modifications to the program which will allow the routine to be relocated to begin at 1FE8 Hex or 8168 decimal. The user must set memory size to 4050 decimal for a system with 4K of memory, and 8167 for a 8 K system.



```
160 POKE 11,232 : POKE 12, 31
```

```
8000 FOR Q = 8168 TO 8191
```

```
8050 DATA 31,136,208,244,169,208
```

```
8060 DATA 141,240,31,96
```

```

20045 POKE 53575,175:POKE 53606,162:
      POKE 53574,154
20050 FOR A=53513 TO 53641 STEP 32
20060 POKE A,161:NEXT A
20065 POKE 53545,32
20070 FOR A=53579 TO 53643 STEP 32
20080 POKE A,161:NEXT A
20090 FOR A=53581 TO 53709 STEP 32
20100 POKE A,161:NEXT A
20110 POKE 53580,161:POKE 53644,161:
      POKE 53708,161
20140 FOR A=53519 TO 53647 STEP 32
20150 POKE A,161:NEXT A
20160 FOR A=53585 TO 53649 STEP 32
20170 POKE A,161:NEXT A
20180 POKE 53584,161
20190 FOR A=53524 TO 53652 STEP 32
20200 POKEA,161:NEXT A
20210 POKE 53523,161:POKE 53525,161
20230 FOR E=1 TO 500:NEXT E
20240 X=USR(X)
20250 RETURN
OK

```

OHIO SCIENTIFIC

HARDWARE: C1P VIDEO—Gives true 32 Chars/Line with guard bands. This is not a make-shift mod. It makes your video every bit as good as the 4P's plus you have switch selectable 1, 2 and 3 MHz. CPU clock as well as 300, 600 and 1200 baud for cassette and serial port all crystal controlled.

Complete Plans — \$18.95 Kit — \$39.95
or Send in your C1P to:

Personal & Business Computer Connection
38437 Grand River
Farmington Hills, MI. 48018

and we will install the video mod. for \$79.95. Other mods available...Add sound: RS-232 port cassette motor control

SOFTWARE: (With Documentation) For C1, C2, 4P & 8P. Chess 1.9, Backgammon, Excellent card games, Arcade type games, Utility Programs, Mini Word Processor, Memory Maps, etc.

Catalog With Free Program (hard copy) and Memory Map for BASIC in ROM Models...\$1.00

Progressive Computing
3336 Avondale Court
Windsor, Ontario, Canada N9E 1X6
(519) 969-2500

NOTE: Any Modification will VOID your OSI warranty.

WHAT'S THE ONE THING NO ONE HAS THOUGHT ABOUT DOING WITH COMPUTERS?



We acknowledge that computers are the most valuable data processing devices ever conceived for business and education, and are the most creative toys on earth. However, the potential of computers has only begun to be explored. Avant-Garde Creations has discovered and developed a way to use computers in the areas of self-transformative experiences, life-awareness, making relationships work, and "getting your act together".

Previously, it was thought that such trips as est, Lifespring, Actualizations, and others were the only means of significantly dealing with the above areas. We acknowledge that they are indeed valuable experiences. But because one has to devote many full days and hundreds of dollars to such trips, all those people who aren't yet ready to get into all this that deeply are left with nowhere to turn for such awareness experiences.

The intention of Avant-Garde Creations is to change that. We have the knowledge and techniques, and now we have the programs. The first is called "The Life Dynamic Transformation Experience", and the second is "The Relationship Life Dynamic". Both are available on disk at this time. Over the next year 9 more life dynamic areas programs will be developed in the following areas: physical, creativity, environment, meaning, conditioning, sexuality, normalcy, aliveness, and responsibility.

Why spend hundreds of dollars to find out if you're into working on such areas of your life? Why not get something you can use over and over, at your convenience?

One of the most exciting aspects of our programs is that in many program sections, it was discovered that "game-playing" was the most effective method of getting people to perceive a difficult truth. So from time to time you're given 100% original games to play, games specifically designed to give you the opportunity to "get" something challenging or elusive. So even if you don't "get" something, you'll have a great time!

Apple II with Applesoft and 48K required with single disk drive.

The Life Dynamic Transformation Experience... \$15.95 postpaid
The Relationship Life Dynamic... \$15.95 postpaid
Complete Program Listings available:
The Life Dynamic Transformation Experience... \$4.95 postpaid
The Relationship Life Dynamic... \$4.95 postpaid



If you don't have an Apple II with Applesoft, there are approximately 5 Apple stores in every state. We'll send you the address of the one nearest you. Your Apple dealer will be glad to give or rent you (very cheaply) some time on one of their computers, and will be happy to get you started running the program. In case you think you need to understand computers to run our programs, we assure you that if you can read English and type your name, you won't have any problem.

AVANT-GARDE CREATIONS

PO Box 30161 Eugene, OR 97403 Dept. C1

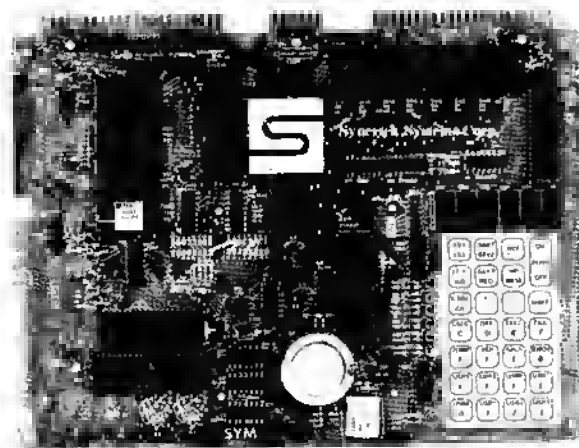


SYM-1, 6502-BASED MICROCOMPUTER

- FULLY-ASSEMBLED AND COMPLETELY INTEGRATED SYSTEM that's ready-to-use
- ALL LSI IC'S ARE IN SOCKETS
- 28 DOUBLE-FUNCTION KEYPAD INCLUDING UP TO 24 "SPECIAL" FUNCTIONS
- EASY-TO-VIEW 6-DIGIT HEX LED DISPLAY
- KIM-1* HARDWARE COMPATIBILITY

The powerful 6502 8-Bit MICROPROCESSOR whose advanced architectural features have made it one of the largest selling "micros" on the market today.

- THREE ON-BOARD PROGRAMMABLE INTERVAL TIMERS available to the user, expandable to five on-board.
- 4K BYTE ROM RESIDENT MONITOR and Operating Programs.
- Single 5 Volt power supply is all that is required.
- 1K BYTES OF 2114 STATIC RAM on-board with sockets provided for immediate expansion to 4K bytes on-board, with total memory expansion to 65,536 bytes.
- USER PROM/ROM: The system is equipped with 3 PROM/ROM expansion sockets for 2316/2332 ROMs or 2716 EPROMs
- ENHANCED SOFTWARE with simplified user interface
- STANDARD INTERFACES INCLUDE:
 - Audio Cassette Recorder Interface with Remote Control (Two modes: 135 Baud KIM-1* compatible, Hi-Speed 1500 Baud)
 - Full duplex 20mA Teletype Interface
 - System Expansion Bus Interface
 - TV Controller Board Interface
 - CRT Compatible Interface (RS-232)
- APPLICATION PORT: 15 Bi-directional TTL Lines for user applications with expansion capability for added lines
- EXPANSION PORT FOR ADD-ON MODULES (5f I/O Lines included in the basic system)
- SEPARATE POWER SUPPLY connector for easy disconnect of the d.c. power
- AUDIBLE RESPONSE KEYPAD



Synertek has enhanced KIM-1* software as well as the hardware. The software has simplified the user interface. The basic SYM-1 system is programmed in machine language. Monitor status is easily accessible, and the monitor gives the keypad user the same full functional capability of the TTY user. The SYM-1 has everything the KIM-1* has to offer, plus so much more that we cannot begin to tell you here. So, if you want to know more, the SYM-1 User Manual is available, separately.

SYM-1 Complete w/manuals	\$229.00
SYM-1 User Manual Only	7.00
SYM-1 Expansion Kit	60.00

Expansion includes 3K of 2114 RAM chips and 1-6522 I/O chip.

SYM-1 Manuals: The well organized documentation package is complete and easy-to-understand.

SYM-1 CAN GROW AS YOU GROW. Its the system to BUILD-ON. Expansion features that are soon to be offered:

*BAS-1 8K Basic ROM (Microsoft)	\$89.00
*KTM-2 TV Interface Board	\$119.00

*We do honor Synertek discount coupons

QUALITY EXPANSION BOARDS DESIGNED SPECIFICALLY FOR KIM-1, SYM-1 & AIM 65

These boards are set up for use with a regulated power supply such as the one below, but, provisions have been made so that you can add onboard regulators for use with an unregulated power supply. But, because of unreliability, we do not recommend the use of onboard regulators. All I.C.'s are socketed for ease of maintenance. All boards carry full 90-day warranty.

All products that we manufacture are designed to meet or exceed industrial standards. All components are first quality and meet full manufacturer's specifications. All this and on extended burn-in is done to reduce the normal percentage of field failures by up to 75%. To you, this means the chance of inconvenience and lost time due to a failure is very rare; but, if it should happen, we guarantee a turn-around time of less than forty-eight hours for repair.

Our money back guarantee: If, for any reason you wish to return any board that you have purchased directly from us within ten (10) days after receipt, complete, in original condition, and in original shipping carton; we will give you a complete credit or refund less a \$10.00 restocking charge per board.

VAK-1 8-SLOT MOTHERBOARD

This motherboard uses the KIM-1* bus structure. It provides eight (8) expansion board sockets with rigid cord cage. Separate jacks for audio cassette, TTY and power supply are provided. Fully buffered bus.

VAK-1 Motherboard	\$139.00
--------------------------	-----------------

VAK-2/4 16K STATIC RAM BOARD

This board using 2114 RAMs is configured in two (2) separately addressable 8K blocks with individual write-protect switches.

VAK-2 16K RAM Board with only 8K of RAM (1/2 populated)	\$239.00
VAK-3 Complete set of chips to expand above board to 16K	\$125.00
VAK-4 Fully populated 16K RAM	\$325.00

VAK-5 2708 EPROM PROGRAMMER

This board requires a +5 VDC and ± 12 VDC, but has a DC to DC

multiplexer so there is no need for an additional power supply. All software is resident in on-board ROM, and has a zero-insertion socket.

VAK-5 2708 EPROM Programmer	\$249.00
------------------------------------	-----------------

VAK-6 EPROM BOARD

This board will hold 8K of 2708 or 2758, or 16K of 2716 or 2516 EPROMs. EPROMs not included.

VAK-6 EPROM Board	\$119.00
--------------------------	-----------------

VAK-7 COMPLETE FLOPPY-DISK SYSTEM (See March Issue of Micro)

VAK-8 PROTOTYPING BOARD

This board allows you to create your own interfaces to plug into the motherboard. Etched circuitry is provided for regulators, address and data bus drivers; with a large area for either wire-wrapped or soldered IC circuitry.

VAK-8 Prototyping Board	\$39.00
--------------------------------	----------------

POWER SUPPLIES

ALL POWER SUPPLIES are totally enclosed with grounded enclosures for safety, AC power cord, and carry a full 2-year warranty.

FULL SYSTEM POWER SUPPLY

This power supply will handle a microcomputer and up to 65K of our VAK-4 RAM. ADDITIONAL FEATURES ARE: Over voltage Protection on 5 volts, fused, AC on/off switch. Equivalent to units selling for \$225.00 or more.

Provides +5 VDC @ 10 Amps & ± 12 VDC @ 1 Amp
VAK-EPS Power Supply

\$119.00

*KIM is a product of MOS Technology

KIM-1* Custom P.S. provides 5 VDC @ 1.2 Amps and +12 VDC @ .1 Amps
KCP-1 Power Supply

\$39.00

SYM-1 Custom P.S. provides 5 VDC @ 1.4 Amps
VCP-1 Power Supply

\$39.00

RNB ENTERPRISES
INCORPORATED

2967 W. Fairmount Avenue
Phoenix, AZ. 85017
(602)265-7564



SYMple BASIC Data Files

The SYM-1 has a Microsoft BASIC available in ROM. Data Save and Data Load via the cassette are NOT supported by this version. The routines required to implement these two important functions are presented here.

John M. Bialock
3054 West Evans Drive
Phoenix, AZ 85023

If you've read "A SYMple Memory Expansion" in the August 1979 issue of MICRO and "Another KIM Expansion" in the September 1979 Issue of *Kilobaud Microcomputing*, then you know that I like Micro-Z's BASIC for the KIM. You will also know that I have the Synertek BAS-1 BASIC for the SYM. Both versions were written by Microsoft, have 9-digit decimal accuracy, etc. but differ in some of their functions.

Comparing the Micro-Z Synertek BASICS

Synertek BASIC has a more convenient USR function and a "&"hex" function that are definite improvements over the original BASIC. Their ROM version has no GET function like Micro-Z's. Another difference is that a response of a carriage return only to an INPUT statement will cause a break in program execution with Synertek's BASIC. Micro-Z has supplied a patch to defeat this break. The Synertek ROM does not include any trig functions, but they have recently released Technical Note #53-SSC that gives you full trig capability using only 313 bytes of RAM.

The main difference between the two BASICS, then, is the data save/data load feature added to his version by Bob Kurtz of Micro-Z. This is a very valuable feature that Microsoft left out. BASIC can not be used to maintain any types of files such as mailing lists, inventory records, or financial records without this feature. Perhaps you could enter the data via DATA statements, but that would be a very trying task indeed! This feature is the major reason that I have preferred Micro-Z's BASIC over Synertek's.

Data Save/Data Load for Synertek BASIC

Listings 1, 2, and 3 are my first attempts to provide the same data save/data load functionality for the SYM with BAS-1. Listing 1 is just BASIC initialization, program loading, and a LIST of the program. All terminal input has been underlined for clarity. The little crooked arrows represent a carriage return typed in.

Listing 2 is a RUN of the program showing the means used to save the data. Three separate records are saved; the page zero pointers, the numeric data and string pointers, and the string data itself. To reload this data, BASIC must be initialized with the same memory size and the program can not have been modified.

Listing 3 is another RUN of the program after memory was cleared and the program reloaded. The data saved in listing 2 was restored, as can be seen. No, it is not as convenient as Bob Kurtz's method, but it works! Bob packs all the data together with a machine language subroutine and save it as one record. Another subroutine loads the combined record and then unpacks it, moving the data back to its original locations.

Machine Language Version of Data Save/Data Load

Listing 4 is a machine language subroutine that will save and load BASIC data files without having to turn control over to the SYM monitor. The data is still saved in three separate records, but they are recorded/loaded one right after another by the routine. An extra few seconds for each save or load (for sync, etc.) shouldn't hurt anyone, should it?

Listing 5 is a VERIFY dump of the subroutine. Load it in, VERIFY between the same addresses, and if you check sums match mine then you keyed it in correctly. Now we know why Synertek put those check sums on the VERIFY dumps! The rest of listing 5 shows BASIC initialization and the loading of the revised BASIC program.

Listing 6 is just a LIST of the revised program. Note the memory size was specified to allow room for the machine language subroutine which is called by statements 100 and 400. With either of the two methods, put the call to the load routine after any DIM statements and before the main program body. The call to the save routine should be at the very end of the program, as shown. Any changes to the program that increases the memory size needed for it will prevent data saved by a prior version from being loaded correctly.

Listing 7 is a RUN of the revised program wherein the data that is entered is saved at the end of the RUN. Listing 8 shows memory being cleared, BASIC initialization identical to that used in listing 7, and then the BASIC program being reloaded. The RUN of the program loads the data saved in listing 7.

If you plan on saving and loading data files very often, dedicating 148 bytes of memory to this subroutine should pay for itself in convenience over the method given earlier.

SYMple Memory Expansion Update

Regular readers of MICRO will recognize from the listings that my SYMple memory expansion board is still work-

Listing 4

```

1 ;
2 ; *** SAVER ***
3 ;
4 ;
5 ; Routine to save and load SYM BASIC data tables.
6 ;
7 ; Initialize BASIC with MEMORY SIZE = 8043 for an 8K SYM.
8 ; Call data load routine with *Q = USR(8136+384)* after
9 ; any DIM statements, but before any processing.
10 ; Call data save routine with *Q = USR(8044+384)* after
11 ; all data processing in the program has been done.
12 ; Always initialize BASIC with the same values and don't
13 ; alter the program or the data will not load properly.
14 ; The code is completely relocatable, only the MEMORY SIZE
15 ; and USR addresses must be changed for other locations.
16 ;
17 ;
18 ; Written by John M. Rialock, August 24, 1979
19 ;
20 ;
21 ; Routine and pointer addresses
22 ;
23 SADR: equ $A64C ; tape starting address
24 EADR: equ $A64A ; tape ending address + 1
25 ID: equ $A64E ; tape record identifier
26 SOD: equ $7D ; beginning of date
27 EOD: equ $81 ; end of date + 1
28 BOS: equ $83 ; beginning of strings
29 EOS: equ $87 ; end of strings + 1
30 NXL: equ $D3 ; next line pointer
31 TMP: equ $EE ; temporary data
32 DUMPT: equ $8E67 ; SYM tape save routine
33 LOADT: equ $8C78 ; SYM tape load routine
34 SAVER: equ $8188 ; SYM register save routine
35 RESALL: equ $81C4 ; restores registers & returns
36 OUTCHK: equ $8A47 ; SYM terminal output routine
37 ;
38 ; Data save routine
39 ;
40 org
41 SADR: $1F6C
42 SAVER: ID
43 ID: STA
44 LDA
45 STA SADR+1
46 STA EADR+1
47 LDA $65
48 STA SADR
49 LDA $EA
50 STA EADR
51 JSR DUMPT
52 JSR IDT
53 LDY $80
54 LDA ROT
55 STA SADR
56 STA ROT+1
57 LDA EOD
58 STA SADR+1
59 STA EOD+1
60 STA EADR+1
61 INC ID
62 JSR DUMPT
63 JSR ROT
64 LDY $80
65 STA ROS
66 STA SADR
67 STA ROS+1
68 LDA STA
69 LDA EOS
70 STA FAD
71 LDA FOS+1

```

Listing 5

```

.V 1F6C-1FFF
1F6C 20 88 81 8D 4E A6 A9 00 53
1F7A 8D 4D A6 8D 4B A6 A9 65 5F
1F7C 8D 4C A6 A9 EA 8D 4A A6 EE
1F84 20 87 8E 20 FA 1F A0 80 7C
1F8C A5 7D 8D 4C A6 A5 7E 8D CD
1F94 4D A6 A5 81 8D 4A A6 A5 08
1F9C 82 8D 4B A6 EE 4E A6 20 0A
1FA4 87 8E 20 FA 1F A0 80 A5 1D
1FAC 83 8D 4C A6 A5 84 8D 4D 22
1FBA A6 A5 87 8D 4A A6 A5 88 9E
1FBC 8D 4B A6 EE 4E A6 20 87 A5
1FCA 8E 4C C4 81 20 88 81 8D 7A
1FCC 4E A6 A5 D3 85 EE A5 D4 D2
1FDA 85 EF 20 78 8C 20 FA 1F A3
1FDC A0 80 EE 4E A6 20 78 8C C9
1FE4 20 FA 1F A0 80 EE 4E A6 04
1FEC 20 78 8C A5 EF 85 D3 A5 B8
1FF4 EF 85 D4 4C C4 81 A9 2A 64
1FFC 20 47 8A 60 B5
4CD5

```

Listing 6

LIST

```

10 REM SYM DATA SAVE/LOAD DEMO PROGRAM
20 REM JOHN BLALOCK AUGUST 24, 1979
30 DIM A(100), B%(100):N = 0
40 PRINT CHR$(12):FOR I = 1 TO 200:NEXT I
50 PRINTTAB(30):DATA SAVE/LOAD DEMO:PRINT:PRINT:PRINT
60 INPUT "DO YOU WANT TO RESTORE PRIOR SAVED DATA? ";Q$
80 IF LEFT$(Q$,1) <> "Y" THEN 200
90 PRINT:PRINT:START RECORDER ON PLAYBACK:PRINT
100 Q = USR(8136,384)
110 PRINT:PRINT:DATA LOADED:PRINT
120 FOR I = 1 TO 100:NEXT I
200 PRINT CHR$(12):FOR I = 1 TO 200:NEXT I:PRINT
210 PRINT "ENTER PAY NUMBER AND NAME, SEPARATED BY A COMMA:"
220 PRINT "FOR EXAMPLE, '12345,JOHN SMITH'"
230 PRINT "ENTER A NEGATIVE NUMBER (-1) AS A PAY NUMBER TO END
ENTRY."
240 PRINT
250 FOR I = N+1 TO 100
260 INPUT A(I),B%(I)
270 IF A(I) < 0 THEN N = I-1:GOTO 300
280 NEXT I:PRINT:TABLE IS FULL!
290 N = I
300 PRINT:THE TABLE NOW CONTAINS THE FOLLOWING DATA:
310 PRINT " # PAY NUMBER NAME"
320 FOR I = 1 TO N
330 PRINT I TAB(11) A(I) TAB(27) B%(I)
340 NEXT I
350 PRINT:PRINT:PRINT
360 INPUT "DO YOU WANT TO SAVE THIS DATA? ";Q$
370 IF LEFT$(Q$,1) <> "Y" THEN PRINT:PRINT:PROGRAM ENDS:END
380 PRINT:PRINT:PRINT
390 INPUT "START RECORDER ON RECORD, THEN ENTER 'G CR', ";Q$
400 Q = USR(8044,384)
410 PRINT:PRINT:DATA SAVED:PRINT:PRINT:PROGRAM ENDS:END
OK

```

Listing 7

```

RUN
DATA SAVE/LOAD DEMO
DO YOU WANT TO RESTORE PRIOR SAVED DATA? NO

ENTER PAY NUMBER AND NAME, SEPARATED BY A COMMA:
FOR EXAMPLE, '12345,JOHN SMITH'
ENTER A NEGATIVE NUMBER (-1) AS A PAY NUMBER TO END ENTRY.

? 12345-JOHN SMITH
? 23456-JANE JONES
? 34567-MARY JOHNSON
? -1.

THE TABLE NOW CONTAINS THE FOLLOWING DATA:
# PAY NUMBER NAME
1 12345 JOHN SMITH
2 23456 JANE JONES
3 34567 MARY JOHNSON

```

```

DO YOU WANT TO SAVE THIS DATA? YES
START RECORDER ON RECORD, THEN ENTER 'G CR'. G
**
DATA SAVED.
PROGRAM ENDS.

.F 00-00-F0
.F 00-200-FFFF
.L2 01
.J 0
MEMORY SIZE? 8043
WIDTH?
7530 BYTES FREE
BASIC V1.1
COPYRIGHT 1978 SYNERTEK SYSTEMS CORP.

OK
LOAD
LOADED
OK
RUN
DATA SAVE/LOAD DEMO

DO YOU WANT TO RESTORE PRIOR SAVED DATA? YES
START RECORDER ON PLAYBACK.
**
DATA LOADED.
ENTER PAY NUMBER AND NAME, SEPARATED BY A COMMA:
FOR EXAMPLE, '12345,JOHN SMITH'
ENTER A NEGATIVE NUMBER (-1) AS A PAY NUMBER TO END ENTRY.

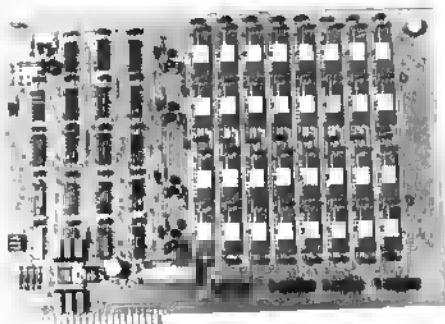
? 44444,JAMES BOND
? 44444,ALLEN SMITH
? -1.

THE TABLE NOW CONTAINS THE FOLLOWING DATA:
# PAY NUMBER NAME
1 12345 JOHN SMITH
2 23456 JANE JONES
3 34567 MARY JOHNSON
4 44444 JAMES BOND
5 44444 ALLEN SMITH

DO YOU WANT TO SAVE THIS DATA? NO
PROGRAM ENDS.
OK

```

16K MEMORY



K-1016

- ADDRESSED AS CONTIGUOUS 16K STARTING AT ANY 8K BOUNDARY
- LOW POWER — 1.6 WATTS TOTAL
- K-1016A — \$340 6 MONTH WARRANTY

SYSTEM EXPANSION



K-1012

- 12 PROM SOCKETS — 2708/TMS 2716, USES THE POWER OF ONLY 1 PROM.
- 32 BIDIRECTIONAL I/O LINES
- FULL RS-232 ASYNC SERIAL COMMUNICATIONS, 75-4800 BAUD
- PROM PROGRAMMER
- K-1012A — \$295

EXPANSION FOR YOUR 6502 COMPUTER

HIGH RESOLUTION GRAPHICS



- 320 x 200 BIT MAPPED GRAPHICS
- 8K RAM AVAILABLE FOR USE
- EACH POINT INDIVIDUALLY ADDRESSABLE
- K-1008A — \$240, PET — \$243 (PLUS PET INTERFACE)

MULTI-HARMONIC 4 VOICE MUSIC



K-1002-2

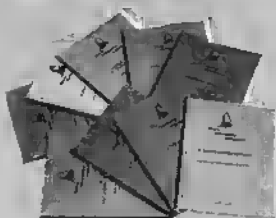


K-1002

MODEL FOR ALL PETS

MODEL FOR KIM, AIM, SYM

- FORIER SYNTHESIZED WAVEFORMS — UP TO 16 HARMONICS
- 4 VOICES PLAY SIMULTANEOUSLY
- QUALITY D/A CONVERTER, 6 POLE FILTER AND AMPLIFIER
- HARDWARE — \$40-50, SOFTWARE — \$20



ALL MTU PRODUCTS ARE SUPPLIED WITH FULL DOCUMENTATION CLASSIFIED AS "BEST IN THE INDUSTRY". MANUALS MAY BE PURCHASED SEPARATELY.



Micro Technology Unlimited

P.O. Box 4596, 841 Galaxy Way
Manchester, N.H. 03108
603-627-1464

Call Or Write For Our Full Line Catalog

A Perpetual Calendar Printer for the AIM

If you know the proper tricks, a Perpetual Calander is quite easy to program. Here it is presented for the AIM 65. In addition to being an interesting demonstration, it points out a few programming tricks required when using integer numbers in BASIC.

Mei Evans
1027 Redeemer
Ann Arbor, MI 48103

Another calendar printer? Yes, but with a couple of new twists. First, it puts out to the AIM printer. So the next time someone asks, "Okay, but what can it actually do?," you can give him an answer he can put in his pocket and take home with him.

Second, it has a built-in perpetual-calander algorithm that finds the starting day-of-the-week for any month of any year from 1583 AD (the start of the Gregorian calendar) to 999999999 AD (or until we change the calendar, or until the world ends, whichever comes first.) The algorithm is fairly simple, but the results can be impressive. For example:

```
RUN
HOW MANY MONTHS? 1
MONTH #? 7
YEAR? 1776
```

```
***** JULY 1776 ***
S M T W T F S
      1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

"So, Independence Day happened on a Thursday."

"You mean it figured out all those leap years clear back to 1776?"

"Well, the equivalent of that, yes."

"How do I know it's right?"

"You don't."

"Okay, print me December, 1941. I know what day Pearl Harbor happened on."

```
RUN
HOW MANY MONTHS? 1
MONTH #? 12
YEAR? 1941
```

```
** DECEMBER 1941 ***
S M T W T F S
      1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

"So December 7th was a Sunday."
"Hey, that's right! Okay, print me the start of year 2000."

```
RUN
HOW MANY MONTHS? 2
FIRST MONTH #? 1
YEAR? 2000
```

```
*** JANUARY 2000 ***
S M T W T F S
              1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

```
** FEBRUARY 2000 ***
S M T W T F S
              1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29
```

"How about that! It got February right. Century years aren't normally leap years, but every fourth century is, and there it is."

"Right. Want a calendar of this month, and maybe the rest of the year?"

"Sure, but make it through next February. Why do all calendars end at December?"

"I don't know, but this one won't."

```
RUN
HOW MANY MONTHS? 5
FIRST MONTH #? 10
YEAR? 1979
```

```
*** OCTOBER 1979 ***
S M T W T F S
      1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

```
** NOVEMBER 1979 ***
S M T W T F S
              1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

```
** DECEMBER 1979 ***
S M T W T F S
              1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

```

*** JANUARY 1980 ***
  S M T W T F S
    1 2 3 4 5
  6 7 8 9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28 29 30 31

```

```

** FEBRUARY 1980 **
  S M T W T F S
    1 2
  3 4 5 6 7 8 9
 10 11 12 13 14 15 16
 17 18 19 20 21 22 23
 24 25 26 27 28 29

```

The day-of-the-week algorithm appeared in **BYTE** (Day of Week and Elapsed Time Programs," W. B. Agocs, **BYTE**, September, 1979, p. 126). I read it, thought "That's neat," and forgot it. Then a calendar printing program for Teletype came out in **Kilobaud** ("Calendar Program," Steve Tabler, **Kilobaud Microcomputing**, October 1979, p. 102). Can the AIM do that on its printer? Sure it can! Can I build in that day-of-week algorithm so that it doesn't need starting instructions? Sure I can! The resulting AIM BASIC program is listed in Figure 1.

The starting day-of-week algorithm is in lines 85 through 150. It uses "Zeller's congruence," as explained in Agoc's article. Zeller first does some juggling of month and year numbers before getting down to the main computation of the day-of-week (variable DW in line 150).

The algorithm packs more power than I needed here; it works for any year, month, and day-of-month (day-of-month is variable DM in line 130). Since I only needed the beginning day-of-week of each month to be printed, I set DM = 1 in line 129. To restore the algorithm to its full power, just delete that one statement, and use DM as an input.

AIM BASIC (like most BASICs) does not allow much format flexibility in printing numbers, so to squeeze those date-lines onto the 20-column printer, a string variable, L\$, is used to build each line before printing. L\$ is first null'd (e.g., line 290), and is then built up, character by character, as in line 350:

```
L$ = L$ + CHR$(48 + D2)
```

This statement adds D2, the second (units) digit of a two-digit date number, to line L\$. As shown in Appendix E of the AIM BASIC manual, CHR\$(48) is ASCII "0" (zero), and the other digits follow. So, if D2 = 5, say, ASCII "5" is added to the string. After the last character has been added, the line is printed (e.g., line 380).

If you are fussy about format, the above technique gives you total control over each column of each line. If numbers don't print to suit you; don't print numbers, print characters.

AIM BASIC has one quirk which I haven't noticed in others (but if you're running a different BASIC, you might like to check it out). If X evaluates internally as less than an integer, but is sufficiently close to that integer, it will print as the integer, but INT(X) will truncate down to the next-lower integer; e.g., if X = 4.99999..., you get:

```

PRINT X
5
PRINT INT(X)
4

```

Don't believe it? Try this:

```

LIST
20 X=5
30 Y=X/3
40 Z=Y*X/3
50 W=3*3*Z/X
60 PRINT "X=";X;" I
INT(X)=";INT(X)
70 END

```

```

RUN
X= 5 INT(X)= 4

```

To prevent this from happening, add a dab to X before doing INT(X). How much is a dab? Anything less than the smallest meaningful increment in X. The first equation in line 258, for example, is computing the century from the year:

```
C = INT(Y/100 + .005)
```

If year Y increases by 1, Y/100 increases by .01, so the added dab is half that. This assures that it will work for the year 2000, and is small enough so it will also work for 1999.

Another example is on Line 262:

```
INT(YC/4 + .1).
```

When YC increments by one, YC/4 increases by .25, and the added dab is less than half that. The previous .005 would work fine here, too, but .1 costs fewer bytes.

A final note of minor interest. Line 80 sends two line-feeds to the printer before starting the calendar, and line 430 sends it five line-feeds, so you can tear off the finished calendar without having to pump the "LF" key. And PRINT TAB(100) is sure neater than a string of five PRINT statements, isn't it?

```

LIST
4 REM
5 REM PERPETUAL-
  CALENDAR PRINTER
6 REM
10 DIM A(12);R$(12)
20 FOR I=1 TO 12:RE
AD A(I):NEXT I
30 FOR I=1 TO 12:RE
AD R$(I):NEXT I
40 INPUT "HOW MANY
MONTHS";N
50 IF N=1 THEN INPUT
T "MONTH #";M
60 IF N>1 THEN INPUT
T "FIRST MONTH #";M
70 INPUT "YEAR";Y
80 PRINT TAB(40)
85 REM CONVERT TO
ZELLER MONTH & YEAR
90 MZ=M-2:YZ=Y
100 IF M=1 THEN MZ=
11:YZ=Y-1
110 IF M=2 THEN MZ=
12:YZ=Y-1
115 REM FIND
STARTING DAY-OF-WEEK
120 CZ=INT(YZ/100+.
005):YZ=YZ-100+CZ:DM
=1
130 D1=INT(.2.6*MZ+.
1)+DM+YZ
140 D1=D1+INT(YZ/4+.
1)+INT(CZ/4+.1)-2*C
Z
150 DW=D1-7*INT(D1
7+.91)+1
155 REM PRINT HEADE
R
160 PRINT R$(M);PR
INT Y;PRINT "***"
170 PRINT " S M T
W T F S"
175 REM BUILD FIRST
DATE-LINE & PRINT
180 L$="":D1=DW-.5
190 FOR I=1 TO 7
200 DT=I-DW+1
210 IF I<D1 THEN L$
=L$+" "
220 IF I>D1 THEN L$
=L$+" "+CHR$(48+DT)
230 IF I<6.5 THEN L
$=L$+" "

```



```

240 NEXT I
250 PRINT L$
255 REM CHECK FOR
    LEAP-YEAR
258 C=INT(Y/100+ .00
5):YC=Y-100*C
260 A(2)=28
262 IF YC=4*INT(YC/
4+.1) THEN A(2)=29
264 IF YC<.5 THEN A
(2)=28
270 IF YC<.5 AND C=
4*INT(C/4+.1) THEN A
(2)=29
275 REM BUILD
REMAINING DATE-LINES
AND PRINT
280 EN=0
290 L$=" "
300 FOR I=1 TO 7

```

```

310 DT=DT+1:IF DT>A
(M)+.5 THEN EN=1:GOT
O 380
320 D1=INT(DT/10+.0
5):D2=DT-10*D1
330 IF D1<.5 THEN L
$=L$+" "
340 IF D1>.5 THEN L
$=L$+CHR$(48+D1)
350 L$=L$+CHR$(48+D
2)
360 IF I<.5 THEN L
$=L$+" "
370 NEXT I
380 PRINT L$
390 IF EN<.5 THEN 2
90
400 PRINT " "
405 REM DO AGAIN
FOR NEXT MONTH

```

```

410 M=M+1:IF M>12.5
THEN M=1:Y=Y+1
420 N=N-1:IF N>.5 T
HEN 90
430 PRINT TAB(100)
440 END
450 REM DATA: MONTH
LENGTHS AND NAMES
460 DATA 31,28,31,3
0,31,30,31,31,30,31,
30,31
470 DATA *** JANUAR
Y,** FEBRUARY,**** M
ARCH
480 DATA **** APRIL
***** MAY ,*****
JUNE
490 DATA ***** JULY
,**** AUGUST,* SEPT
EMBER
500 DATA *** OCTOBE
R,** NOVEMBER,** DEC
EMBER

```

AIM 65 Software

NEW!

*** DISCOVER 6502 POWER ***

HELP!!

9 Super utility programs for all AIM 65 programmers. **HEX INPUT:** Long and short versions, used for entering hex bytes into memory. **DUMP & HEX OUT:** Print out your memory in two formats for easy checking or location of individual bytes. **FIELD SORT:** A field sorting routine that finds usages of many tasks including helping you organize your programming. **RESTORE:** A program which automatically restores your editor after you've re-entered it improperly. This has been a real time saver for us. **DNE STEP:** Allow you to step thru the disassembly (listing) one line at a time. **SYMBOL TABLE:** Is for use with the disassembler ROM (How can you do without one?) It prints the beginning and ending addresses of your symbol table along with each label in your program and its address, all in a handy format. **RELOCATE:** Is a powerful program which allows you to move or relocate programs or data in memory. All who write, adapt or create programs or subroutines will appreciate this. It allows you to place them wherever you'd like. You can even open up space right in the middle of a program for inserting missing new or additional code or instructions. A programmer's dream.

GAIMS PAK I

5 Exciting games of skill for 1 or several players, using the full capabilities of the AIM 65 keyboard, display, and printer. **HANGMAN:** An engrossing word game for 2 players. The AIM does the work and keeps score. **SCORE 4:** An engrossing game in 3 dimensions. The printer shows the positions of the 2 players after each move. **REACT:** You reflexes are tested in hundredths of a second. The display and keyboard are turned into a reaction timer. **GOL-LUMS CAVERNS:** Place you into the underground kingdom of the evil Wizard. You must move thru secret tunnels and cavern rooms avoiding traps, mysterious mists, and the Wizard's spell. To capture the Wizard you have only a few poison darts and your Magic computer to warn you when Evil is near. **BINHEX:** Teaches and tests your ability to convert Binary numbers into their Hexadecimal equivalent. Fun for budding programmers and helps you to perfect a needed skill for fast and efficient programming.

MATH WHIZ

6 Programs dealing with numbers & math. **AIM 65 ADD & SUBTRACT:** This powerful utility program turns your AIM 65 into a multiple precision calculator. **TOTAL:** Adds up to four decimal or hexadecimal numbers at a time. **TEST MEMORY:** Lets you really check out your RAM memory. **FIBBONACCI:** You learn about these important numbers as your AIM generates them in a series. **DEC TO HEX:** A multi-use program and algorithm for changing decimal numbers into their hex equivalents. **TIMER:** Makes your AIM 65 into a timer or a 12 or 24 hour clock, displaying of printing hours, minutes and seconds. A superb demo of the power of the AIM 65.

YOU CAN NOW DEMONSTRATE THE POWER OF YOUR AIM 65 WITH CYBERDYNE'S DYNAMIC ACTION SOFTWARE. ALL OF OUR PROGRAMS RUN ON 1K OR 4K AIMS. ALL SOFTWARE ON EZ-LOAD TAPE CASSETTES. COMPLETE TEXT, PROGRAM, & LOADING INSTRUCTIONS ARE INCLUDED.

"TAKE AIM" MANUAL, VOL I by JAMES HDYT CLARK of CYBERDYNE'S staff, coming soon—watch for it or write for info. A guide for all. Master AIM 65 hardware & software. A tab and learning manual, extension, clarification, & index to AIM 65 documentation. Over 30 programs. Explained & fully documented (games, math, utility, printer, display, & more).

HOT LINE!!

(801) 224-2745

GAIMS PAK II

6 Value packed games of skill and chance at less than \$1.75 each. Created for maximum enjoyment of the AIM 65 by you and others. **BRICKS:** This program is unique because it learns from your mistakes and guesses, while you play. It actually becomes "smarter" as you and the computer compete in a series of games. A real challenge to your skills of logic, deduction, and memory. **TIC-TAC-TOE:** Need we say more than the AIM 65 is a fair and impartial scorekeeper. **CARDS:** Gives you practice in when to hold on and when to fold on a your AIM 65 double CARD STUD from an unmarked and randomly shuffled deck. **LOGICAL ORDER:** Test your skills as a Master-Mind of reason and logic as you try to deduce a random 4 number sequence in the fewest number of tries. **STARWAY 090:** Places you at the controls of a crippled spacecraft. You must successfully pilot your craft back to the mother ship for a soft rendezvous. Your supply of fuel is limited and must be used with care to avoid disaster. **ESP:** Even computer geniuses ESP (or seem to). You mentally pick a number, answer a few questions (without disclosing the number), and your AIM 65 will guess the number correctly every time.

SHOW OFF

7 Programs (less than \$1.50 each) which show off all the features of the AIM 65. **SIGNS:** Lets you print 2 sizes of letters edge-wise on the AIM 65 printer & make it go bannet. **ROTATING BILLBOARD:** Show your messages as they rotate along the display. **PRINTER WAVE:** A good demo of user control of the AIM 65 printer. Store you into graphics printing. **PAPER ADVANCE:** Give you full control of the printer paper advance. See those last few lines printed for exchange. **LINE FEED:** The time it is hard to control of the paper advance. **CURSORS DEMO:** You can light all 16 segments of the display or type in different display patterns. **KEYBOARD INTERRUPT:** Gives you hardware control of the keyboard. Sends the keyboard to key enclosure without interrupting program execution. ALL PROGRAMS in this section may be used in your own programs as subroutines or run on their own as demos. Full instructions included.

AIM 65 & 6502 RELATED PRODUCTS

I/O-TTY-CASSETTE connector board for the AIM 65. Plug directly to AIM 65 app connector (J1). Includes AIM connector, TTY and recorder (cable col for I recorder, PC board with traces & holes for LED indicators, switch input sensors optoisolators drivers, relays, and amp. & AIM 65 I/O training course with software. AIM I/O BOARD-KIT \$19.75. Assembled \$22.75.

CYBERDYNE'S 1980 CATALOG G. HARDWARE. SOFTWARE BOOKS. TRAINING ITEMS. GOODIES & RAD SERVICES. (FULL DESCRIPTIONS) \$1.00 (PEFUNDABLE ON FIRST ORDER)—FREE WITH ORDER.

ORDER NOW!

ORDER BY ITEM NAME AND PRICE. SOFTWARE CASSETTES \$9.75 EACH. POSTAGE IN U.S.A. & CANADA 25¢ PER ITEM. OUTSIDE U.S. ADD 10% FOR AIR POSTAGE & HANDLING. FOR RUSH ORDERS PAY BY PDSTAL MONEY ORDER. 10% DISCOUNT ON ORDERS OF 3 OR MORE ITEMS. DEALER INQUIRIES INVITED. CUSTOM INDUSTRIAL MICRO-SYSTEMS OUR SPECIALTY.

*** SATISFACTION GUARANTEED ***



Cyberdyne

P.O. Box 1285
Orem, Utah 84057



Introducing AppleSeed, our newest publication to whet your Apple* appetite!

We invite you to subscribe to AppleSeed - the magazine that is to the Apple II* what SoftSide is to the TRS-80**. It offers the newest in software programming hints and ideas tailored especially for your computer. AppleSeed features challenging programs for both the do-it-yourselfer and the individual interested in pre-packaged programs and games... your own preview of the best available on the market today. A typical slice of AppleSeed consists of one major (new 18K) commercial level program (completely listed for your keying pleasure), accompanied by two or three applications for practical use or fun, supplemented by informative articles to polish your Apple*. Get right to the core of your Apple* needs and order AppleSeed today! 12 issues, 1 year, \$15.00. AppleSeed is the newest member of...

SoftSide™

PUBLICATIONS

6 South Street, Milford, NH 03055
(603) 673-5144

*A registered trademark of Apple Computers. **A registered trademark of Radio Shack and Tandy Corp.

Bi-Directional Scrolling

Everyone knows that a teletype only moves the paper in one direction - up. Likewise, the Apple display only scrolls one way - up. Now you can have scrolling in both directions - up and down - with these routines..

Roger Wagner
SW Data Systems
P.O. Box 582
Santee, CA 92071

By using the machine language routines given below, it is possible to scroll either text/gr page in either direction.

The up-scroll routine is derived from APPLE computer's red Reference Manual with the difference being that a zero-page location is referred to determine which page to scroll. The down scroll routine makes similar use of the same zero-page byte.

To use the routine a few entry conditions must be met:

1. Load the binary routine into the \$300 page of memory starting at \$300.
2. Set pointers 6,7, and 8,9. If you want to bring new information onto the screen from RAM as you scroll 6,7 must point to the location in memory where the data to be loaded onto the top line of the screen will come from when you scroll the screen page down. Similarly 8,9 point to the place in memory to get the data for the bottom line when you scroll up.

If you want to use this routine to directly view memory, the easiest way to set the pointers 6,7 and 8,9 is to set 8 and 9 to the address you want to start viewing at. Put the low order byte in 8 and the high order in 9. (The screen height plus 1.) Then set 6,7 to the same value as 8,9 were originally, i.e., the low and high byte bring the starting address. Last of all, scroll back down one line to bring the starting address line into position as the first line of text visible at the top of the screen.

If you do not want new data brought onto the screen, then 6,7 and 8,9 will have

```
10 LOMEM=3072
20 REM OR SET LOMEM MANUALLY BEFORE RUNNING.
30 CALL -936: INPUT "PAGE 1 OR 2?":PAGE
40 PRINT "INPUT ADDRESS (< 32767) TO START AT:" INPUT A
50 REM TO SCROLL WITHOUT BRINGING IN NEW DATA ENTER '0' FOR ADDRESS.
60 IF A#0 THEN 100: TEXT : CALL -936: POKE 34,1: REM FREEZE ONE BLANK LINE AT TOP OF SCREEN
70 YTAB 12: PRINT "(SAMPLE PG.1 SCREEN DATA)"
80 POKE 6,0: POKE 7,4: POKE 8,0: POKE 9,4: REM BRING NEW SCREEN DATA FROM THAT BLANK LINE
90 GOTO 150
100 LB=A MOD 256:HB=A/256
110 POKE 5,PAGE*4: IF PAGE=2 THEN POKE -16299,0
120 POKE 8,LB: POKE 9,HB
130 FOR I=1 TO 25: CALL 768: NEXT I
140 POKE 6,LB: POKE 7,HB
150 KEY= PEEK (-16384): POKE -16368,0
160 IF KEY=149 THEN CALL 768: REM RT. ARROW KEY TO SCROLL UP
170 IF KEY=136 THEN CALL 845: REM LFT. ARROW KEY TO SCROLL DOWN
180 IF KEY#136 AND KEY#149 OR A#0 THEN 190: POKE 6,0: POKE 7,4: POKE 8,0: POKE 9,4: REM RESET 6,7 & 8,9 TO POINT AT BLANK LINE
190 IF KEY#177 THEN 200: POKE 5,4: POKE -16300,0: REM '1' FOR PAGE 1
200 IF KEY#178 THEN 210: POKE 5,8: POKE -16299,0: REM '2' FOR PAGE 2
210 IF KEY#216 THEN 150: POKE -16300,0: TEXT : CALL -868: PRINT "BYE.": END
```

1 *****	0300 A5 22	44 SCROLL LDR WNDTOP	0362 E9 00	99 SBC #00
2 *	0302 48	45 PHA	0364 C5 22	100 CMP WNDTOP
3 * APPLE SCROLLING ROUTINE *	0303 20 9E 03	46 JSR VTABZ	0366 30 00	101 BMT LDTOP
4 *	0306 A5 28	47 NXTLN LDR BASL	0368 48	102 PHA
5 * BY *	0308 85 2A	48 STA BAS2L	0369 20 9E 03	103 JSR VTABZ
6 * ROGER WAGNER *	030A A5 29	49 LDA BASH	036C B1 28	104 NXTCHR2 LDA (BASL),Y
7 *	030C 85 28	50 STA BAS2H	036E 91 2A	105 STA (BAS2L),Y
8 * THIS WILL LET EITHER PAGE *	030E A4 21	51 LDY WNDWOTH	0370 88	106 DEY
9 * SCROLL IN EITHER DIRECTION. *	0310 88	52 DEY	0371 10 F9	107 BPL NXTCHR2
10 * IT IS PRIMARILY DESIGNED *	0311 68	53 PLA	0373 30 E1	108 BMT NXTLN2
11 * TO FEED NEW SCREEN DATA IN *	0312 69 01	54 ADC #01	0375 A0 00	109 LDTOP LDY #00
12 * FROM A GIVEN RANGE OF RAM *	0314 C5 23	55 CMP WNDBTM	0377 B1 06	110 LT2 LDA (SCRNTP),Y
13 *	0316 80 00	56 BCS LDBTM	0379 91 28	111 STA (BASL),Y
14 *****	0318 48	57 PHA	037B C8	112 INY
15 *	0319 20 9E 03	58 JSR VTABZ	037C C4 21	113 CPY WNDWOTH
16 *	031C B1 28	59 NXTCHR LDA (BASL),Y	037E 90 F7	114 BCC LT2
17 *	031E 91 2A	60 STA (BAS2L),Y	0380 38	115 CRRCT2 SEC
18 OBJ \$300	0320 88	61 DEY	0381 A5 06	116 LDA SCRNTF
19 ORG \$300	0321 10 F9	62 BPL NXTCHR	0383 E5 21	117 SBC WNDWOTH
20 WNDLFT EQU \$20	0323 30 E1	63 BMT NXTLN	0385 85 06	118 STA SCRNTF
21 WNDWOTH EQU \$21	0325 A0 00	64 LDBTM LDY #00	0387 A5 07	119 LDA SCRNTF+1
22 WNDTOP EQU \$22	0327 B1 00	65 LD2 LDA (SCRNBTH),Y	0389 E9 00	120 SBC #00
23 WNDBTM EQU \$23	0329 91 28	66 STA (BASL),Y	038B 85 07	121 STA SCRNTF+1
24 CH EQU \$24	032B C8	67 INY	038D 38	122 SEC
25 CV EQU \$25	032C C4 21	68 CPY WNDWOTH	038E A5 08	123 LDA SCRNBTH
26 BASL EQU \$28	032E 90 F7	69 BCC LD2	0390 E5 21	124 SBC WNDWOTH
27 BASH EQU \$29	0330 18	70 CRRCT CLC	0392 85 08	125 STA SCRNBTH
28 BAS2L EQU \$2A	0331 A5 06	71 LDA SCRNTF	0394 A5 09	126 LDA SCRNBTH+1
29 BAS2H EQU \$2B	0333 65 21	72 ADC WNDWOTH	0396 E9 00	127 SBC #00
30 PAGE EQU \$05	0335 85 06	73 STA SCRNTF	0398 85 09	128 STA SCRNBTH+1
31 * FOR APPLESOFT USE PAGE EQU \$1F	0337 A5 07	74 LDA SCRNTF+1	039A 60	129 RTS
32 * PAGE MUST HOLD \$04 FOR PG. 1.	0339 69 00	75 ADC #00	039B 00	130 BRK
33 * \$08 FOR PG. 2	033B 85 07	76 STA SCRNTF+1		131 *
34 SCRNTF EQU \$06	033D 18	77 CLC		132 *
35 * \$05,\$07 = LO/HI BYTES	033E A5 08	78 LDA SCRNBTH	039C A5 25	133 VTAB LDA CV
36 * OF START OF LINE JUST BEFORE	0340 65 21	79 ADC WNDWOTH	039E 20 A6 03	134 VTABZ JSR BASCALC
37 * TOP LINE	0342 85 08	80 STA SCRNBTH	03A1 65 20	135 ADC WNDLFT
38 SCRNBTH EQU \$08	0344 A5 09	81 LDA SCRNBTH+1	03A3 85 28	136 STA BASL
39 * \$08,\$09 = LO/HI BYTES	0346 69 00	82 ADC #00	03A5 60	137 RTS
40 * OF START OF LINE JUST AFTER	0348 85 09	83 STA SCRNBTH+1		138 *
41 * BOTTOM LINE	034A 4C 9C 03	84 JMP VTAB		139 *
42 *		85 *	03A6 48	140 BASCALC PHA
43 *		86 *	03A7 4A	141 LSR
	034D 38	87 SCROLLON SEC	03A8 29 03	142 AND #03
	034E A5 23	88 LDA WNDBTM	03AA 05 05	143 ORA PAGE
	0350 E9 01	89 SBC #01	03AC 85 29	144 STA BASH
	0352 48	90 PHA	03AE 68	145 PLA
	0353 20 9E 03	91 JSR VTABZ	03AF 29 18	146 AND #18
	0356 A5 28	92 NXTLN2 LDA BASL	03B1 90 02	147 BCC BSCLC2
	0358 85 2A	93 STA BAS2L	03B3 69 7F	148 ADC #7F
	035A A5 29	94 LDA BASH	03B5 85 28	149 BSCLC2 STA BASL
	035C 85 28	95 STA BAS2H	03B7 0A	150 ASL
	035E A4 21	96 LDY WNDWOTH	03B8 0A	151 ASL
	0360 88	97 DEY	03B9 05 28	152 ORA BASL
	0361 68	98 PLA	03BB 85 28	153 STA BASL
			03BD 60	154 END RTS

— END ASSEMBLY —
TOTAL ERRORS 00

Change of Address?

We are still having problems with those of you who are moving around. Please notify us of any change of address so that you will not miss any issues. The Post Office does not return the undelivered copies, so we lose both the postage and the magazines. Send address changes to MICRO, c/o Carol Stark, Box 6502, Chelmsford, MA, 01824. Please include your old label or your subscription number.

to point to a part of memory that contains 40 blank space characters. One way to do this is to freeze on blank line on either page 1 or 2, and then set 6, 7 and 8, 9 must be reset to that value each time the scroll is done. This is because normally the scroll routine updates 6,7 and 8,9 by the screen width so as to remain synchronized with the screen display another technique is to just clear the top or bot-

tom line to blanks each time a scroll is done.

3. Location 5 must hold a 4 for page 1 scrolling, and an 8 for page 2.
4. That's all. Now when you want the screen to scroll just 'CALL 768' to scroll up, and '845' to scroll down.

Special Notes:

If you are going to use page 2 of text/gr in Integer Basic, be sure to protect the variables with a 'LOMEM': 3072. This may be done before running the program, or if you know how, put as an early line in the program.

43000 38F

```
0300- A5 22 48 20 9E 03 A5 28
0308- 85 2A A5 29 85 2B A4 21
0310- 88 68 69 01 C5 23 8A 00
0318- 48 20 9E 03 B1 28 91 2A
0320- 88 10 F9 30 E1 A0 00 B1
0328- 08 91 28 C8 C4 21 90 F7
0330- 18 A5 06 65 21 85 06 A5
0338- 07 69 00 85 07 18 A5 08
0340- 65 21 85 08 A5 09 69 00
0348- 85 09 4C 9C 03 38 A5 23
0350- E9 01 48 20 9E 03 A5 28
0358- 85 2A A5 29 85 2B A4 21
0360- 88 68 E9 00 C5 22 30 00
0368- 48 20 9E 03 B1 28 91 2A
0370- 88 10 F9 30 E1 A0 00 B1
0378- 06 91 28 C8 C4 21 90 F7
0380- 38 A5 06 E5 21 85 06 A5
0388- 07 E9 00 85 07 38 A5 08
0390- E5 21 85 08 A5 09 E9 00
0398- 85 09 60 00 A5 25 20 A6
03A0- 03 65 28 85 29 60 48 4A
03A8- 29 03 05 05 85 29 68 29
03B0- 18 90 02 69 7F 85 28 0A
03B8- 0A 05 28 85 29 60 FF FF
```

To use page 2 in Applesoft is more difficult, but can be done. First, location \$3AB in the machine code must be changed from \$05 to \$1F. Also, you must POKE 31 with a 4 or 8 as compared to the POKE 5 in Integer.

The real rub is that Applesoft programs normally begin in memory at \$800 (hex) which conflicts with page 2 use. The way around this is to do a 'POKE 104, 12: POKE 3072, 0' before loading your program. After loading do a 'CALL 54514' (unnecessary with DOS 32). Unless you do a 'RESET', 'Control-B' other programs. Unfortunately, use of page 2 with the RAM version of Applesoft is to my knowledge impossible. (Sorry...)

If you wish to move the scrolling routine for some reason, the only location-dependent aspects of the code are 5 'JSR's and 1 'JMP' within it. Since these operations always reference absolute addresses they will have to be rewritten. Of course, if you have a relocate utility, it is that much easier.

For further enlightenment, see the sample Integer Basic program which makes use of the scrolling routine. Have Fun!

Location dependent:

```
$303: JSR $39E
319: JSR 39E
34A: JMP 39C
353: JSR 39E
369: JSR 39E
39E: JSR 3A6
```

If page 2 of TEXT/CR is to be used, it must be protected by a 'LOMEM:3072' for Integer BASIC, or a 'special Load' (as described in article) when using Applesoft.

Note: \$3AB must be changed from \$05 to \$1F for Applesoft.

Symbol Table

```
WINDLFT 0020
WINDWTH 0021
WINDTOP 0022
WINDBTM 0023
CH 0024
CV 0025
BASL 0026
BASH 0029
BAS2L 002A
BAS2H 002B
PAGE 0005
SCRNTP 0006
SCRNBTH 0008
SCROLL 0300
NXTLN 0306
NXTCHR 031C
LDBTM 0325
LD2 0327
CRCT 0330
SCROLLON 0340
NXTLN2 0356
NXTCHR2 036C
LDTOP 0375
LT2 0377
CRCT2 0380
VTAB 039C
VTAB2 039E
BASCOLC 03A6
BSOLC2 03B5
END 0380
```

SYNERGISTIC SOFTWARE

presents

THE MODIFIABLE DATABASE by Chris Anson & Robert Clardy

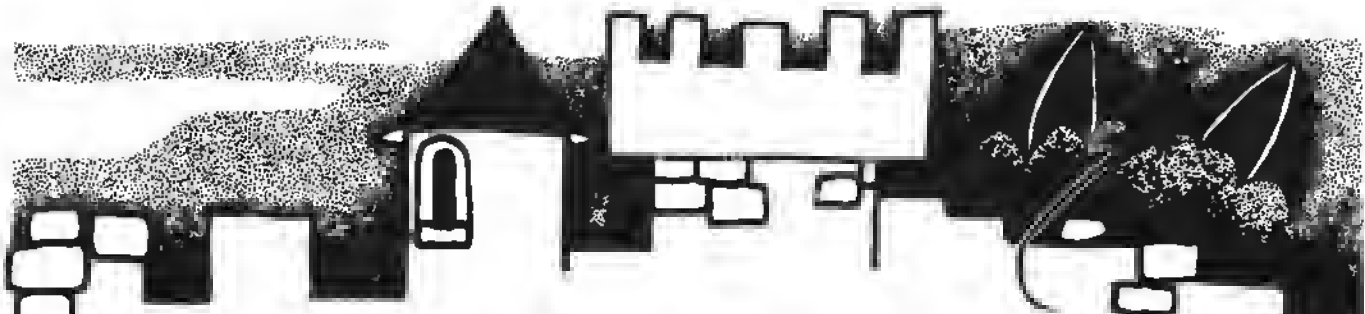
The Modifiable Database is a general purpose, user oriented database program that can be easily customized for your specific data management application. Create any number of application programs such as mailing lists, bibliography files, inventory controls, personnel files, accounting programs, etc. The only limitation is your own imagination.

The program uses fast and flexible machine language search and sort routines, provides for easy record editing, and can search or print up to 2 disks of records with a single command. All commands are invocable by a few keystrokes. There's never been an easier to use or more flexible data management program.

Applesoft program requires 48K and disk	\$49.50
Modifier Module 1 lets you add accounting and numeric processing features to your program	\$15.00
Modifier Module 2 lets you format your output in any way desired (columnar, standard forms, suppressed fields, etc.)	\$15.00



AVAILABLE AT YOUR LOCAL DEALER OR SEND CHECK OR INQUIRY TO SYNERGISTIC SOFTWARE,
5221 120 AVE. S.E., BELLEVUE, WA 98006. WA RESIDENTS ADD 5.3% SALES TAX




Did you read about the Dungeonmaster who became so enchanted playing a real life version of Dungeons and Dragons that he disappeared for a month?

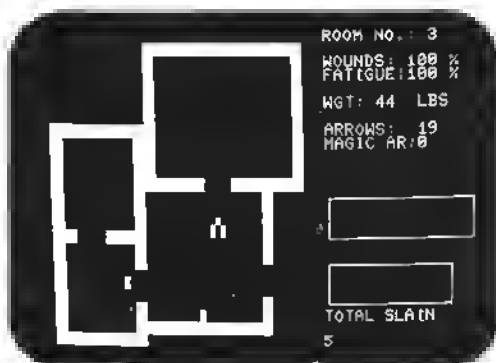
You'll be able to hold on to reality just a little better when you play the **Dunjonquest™** computer version, the greatest of all the role-playing fantasies.

But don't bet on it.

Sit at your computer. You're the hero. Enter the Dunjonquest "Temple of Apshai" and into the greatest fantasy adventure you've ever experienced. The Temple has over 200 rooms and catacombs in which lurk more than 30 kinds of monsters and beasts ready to do you in—in real time—before you can reach any of the 70 or so treasures waiting for the hero. You may spend days, weeks, months... the rest of your life... striking at the forces of evil, or running from them, or calling on powers you can never completely understand. Always, always demonstrating in varying degrees your strength, constitution, dexterity, intelligence, intuition, the force of your ego.



Unlike chess or bridge or monopoly, this role-playing game—like other good role-playing games—is an **experience** rather than a game: it is not played so much as it's lived or experienced. Your alter ego goes forth into the world of demons and darkness, dragons and dwarves. Your character will do whatever you want him (or her or it) to do.



Actual photo of screen during a Dunjonquest game. In Room 3 in the Temple of Apshai, our hero observes two treasures unattended by dragons, monsters or demons... for the moment. He is completely free of wounds; he is not at all fatigued. He carries 44 pounds of armor and 19 arrows in his quiver. He has already slain five demons. Will he capture the treasures before moving on... or before the forces of darkness intercept him?

"The Temple..." comes complete with a superbly illustrated 56-page rule book and cassette program, designed to operate with the Level II 16K TRS 80, the PET 32K or the Apple II 48K (Applesoft) computer. Only \$24.95 complete, **including** shipping and handling on orders placed within the next 30 days. (Apple or TRS 80 disk available for \$29.95).

Dunjonquest's "The Temple of Apshai" is **guaranteed** to be the best version of Dungeons and Dragons/ Dragons and Dungeons. It's a product of the two guys who **are** Automated Simulations: Jim Connelley and Jon Freeman. Jim is a Dungeon Master, running continuous D & D campaigns. He's been a data processing professional with Westinghouse, GTE Sylvania, Logisticon... an expert in computer-based math-modeling and in simulation of complex phenomena. Jon is a game player, designer and author. He's a frequent contributor to *Games* magazine; his book, "The Playboy Winner's Guide to Board Games" is a paperback best-seller.

As we said, **guaranteed**: Guaranteed to be the best version; guaranteed that you'll be happy with it. Order now, use it for two weeks. If you don't enjoy completely this fantasy adventure experience that goes beyond all others, send it back to us. We'll refund your money in full; no questions asked.

Master Charge or Visa card holders: charge "The Temple of Apshai" to your credit card. Just call our toll free number: (800) 824-7888, operator 861 (In California, call operator 861 (800) 852-7777. In Hawaii and Alaska, operator 861 (800) 824-7919) and you can begin enjoying your Dunjonquest game in days. Or send your check for \$24.95* (or \$29.95)* to



**Automated
Simulations**

Dept. LW
P.O. Box 4232
Mountain View,
CA 94040



*California residents, please add 6.5% tax.

Software for the PET

PROGRAMMA

Software Program Products



DOMINOES \$ 6.95



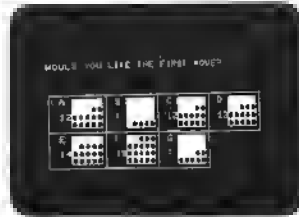
OTHELLO \$ 9.95



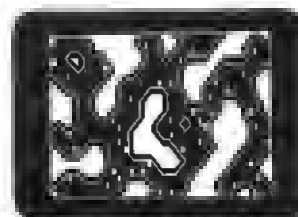
LETTER SQUARES \$ 6.95



RPN MATHPACK \$19.95



SUPER NIM \$ 6.95



DIR/REF \$ 6.95



SPACE WARS \$ 9.95



GRAND PRIX \$ 6.95



E.S.P. \$ 9.95



FORECAST \$ 9.95



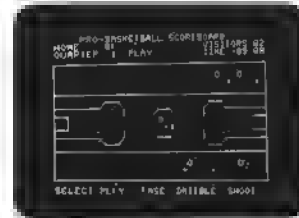
BLOCKADE \$ 9.95



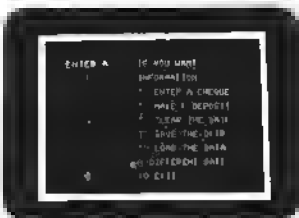
BASKETBALL \$ 9.95



DEPTH CHARGE \$ 9.95



SLOT MACHINE \$ 6.95



HOME ACCOUNTING \$ 9.95

All orders include 3% postage and handling with a minimum of \$1.00. California residents include 6% Sales Tax.

VISA

MASTERCARD

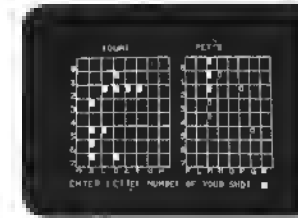
PET IS A TRADEMARK OF
COMMODORE BUSINESS MACHINES, INC.

PROGRAMMA
INTERNATIONAL, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579
384-1116
384-1117

Dealer Inquiries Invited



SHOOTING GALLERY \$ 9.95



BATTLE SHIP \$ 9.95



ZAP \$ 6.95

The SY6516 Pseudo-16 Bit Processor

While the 6502 is a great microprocessor as it stands, advances are being considered to make it even better. One of the approaches is to add some new capabilities such as some 16 bit operations, improved addressing, and more.

Randall Hyde
12804 Magnolia
Chino, CA 91710

For those of you who may have wondered what the 6502 equivalent of the MC6809 would be, wonder no longer. Synertek is almost ready to ship the SY6516.

Synertek announced the 6516 almost a year ago, but due to production problems, it never quite made it. The 6516 was designed by Atari Inc. (back then it was to be called the 6509) for use with the Atari 400 and 800 computer systems. Unfortunately, Synertek was unable to deliver the chip in time for Atari to use it in their computers.

What Is a Pseudo 16-bit Computer?

A pseudo 16-bit computer uses an internal 16-bit register arrangement, but externally it uses an eight bit bus. Sixteen bit data is multiplexed in, much like the Alpha Micro computer on the S-100 bus. In addition to the new 16-bit instructions, the 6516 maintains all of the 8-bit instructions of the 6502. You may reassemble your source files currently on the 6502 and run them directly on the 6516. All the information that I have received says that the 6516 is SOURCE code compatible with the 6502 and that it is OBJECT code incompatible with the 6502. I have heard rumors that Synertek is attempting to make the 6516 object code compatible, but quite honestly, I don't believe there is much chance of it happening.

Unlike the Motorola MC6809, which has a distinct set of 8-bit instructions and a distinct set of 16-bit instructions, the SY6516 contains a special register (the "Q" register) which toggles the system back and forth between 8-bit operation and 16-bit operation. In addition, all registers in the 6516 (A, X, Y, and SP) are

now 16-bits wide. The "Q" register contains four bits which may be programmed to put the accumulator in the 16-bit mode, the X-register in the 16-bit mode, the Y-register in the 16-bit mode, and memory in the 16-bit mode (for use with INC, DEC, ASL, ROL, ROR, LSR, etc.). If the accumulator is programmed to be in the 16-bit mode, then LDA will load the accumulator with 16-bits, the low order byte coming from the specified address and the high order byte coming from the specified address plus one. If the accumulator is in the 8-bit mode, then the LDA instruction behaves identically to the LDA on the 6502. The other registers (X, Y, and Memory) behave identically.

It does not take twice as long to perform a 16-bit instruction compared to the equivalent 8-bit instruction, as you might expect. Usually only one additional clock cycle is required. This means that 6516 code will run as much as 3 times faster than 6502 code performing the same operation.

In addition, several instructions have been "speeded up" over the 6502 equivalent. For instance, implied instructions now only require one cycle for complete execution (the 6502 requires 2). Several other instructions have been speeded up as well (see Table One).

Variety of addressing modes is what makes the 6502 as flexible as it is. The 6516 includes many more addressing modes in its instruction set. In particular, indirect addressing (without the indexed by Y or preindexed by X), 16-bit relative addressing (there is now a jump relative, so your code can be relocatable), and direct page addressing.

Direct page addressing is something

really special. It is available on the 6502 in a restricted form; on the 6502 it is called zero page addressing. Direct page addressing is different, in that any of the 256 pages in the 6516 address space may be used. The particular page is selected by the 8-bit direct page register "Z". The direct page facility should clear up many problems associated with zero page conflicts occurring in the 6502.

The New Instructions

The 6516 has a total of 114 instructions (compared to the 6502's 56). This gives a total of 255 different opcodes. Some of the new instructions are listed on the next page.

The User Flag

Bit 5 of the P register has been undetermined to this point in the 6502. The 6516 utilizes this bit as a user defined flag. Included in the instruction set are instructions to set and clear this flag, as well as branch if set, and branch if clear. This user defined flag will prove to be a great help to users who are writing a boolean function. Up till now, the 6502 programmer had to use the carry or overflow flag. The user defined flag will help alleviate problems associated with the use of the aforementioned flags.

The 6516 instruction set was defined to allow maximum capability with the minimum number of instructions possible. For those of you who would really like to have seen an instruction of the form:

JMP (LBL,X)

you may simulate this by:

LDY LBL,X
YPC

The instruction sequence still requires only 3 bytes (assuming LBL is a direct page reference) and the timing is 7 cycles which is only two cycles more than a straight jump indirect. This would execute just as fast as a JMP (LBL,X) instruction were it included directly in the instruction set.

For those of you who would like to have seen the auto-increment and auto-decrement instructions of the MC6809, once again they can be simulated by the 6516. For instance, the sequence LAX, INX simulates a post increment and INX, LAX simulates a pre-increment. These instructions require two bytes (the same as the 6809) and execute in 3 to 4 cycles (depending on whether you are in the eight-bit or 16-bit mode). This speed is comparable to the 6809.

The only advantage of the 6809 over the 6516 is the 6809 multiply instruction. However, a software multiply on the 6516 should execute fast enough so that it won't make that big a difference.

The addition of two stacks in the 6809 is no real advantage since you can simulate 2, 3 or even n stacks with one 16-bit stack pointer. Those of you writing machine interpreters (such as the UCSD Pascal Pcode interpreter) will be able to simulate a stack machine quite easily on the 6516.

In my opinion, Synertek has taken everything wrong with the 6502 and fixed it, in addition to adding several features which I had not even previously considered. The 6516 is easily the most powerful 8-bit processor available (with due respects to the Intel 8088 which I would rate "almost there"). This opinion, incidentally, is not just my own. EDN rated the 6516 above all the 8-bit processors and even some 16-bit processors, several months ago. If Synertek does indeed make the 6516 processor object code compatible with the 6502, it will definitely make the 6516 something you shouldn't scoff at. Why? Because once this happens, 50,000 APPLE II computers will be upgradeable directly to a 16-bit processor and maintain software compatibility with existing software. Likewise, the 70,000 or so PETs will be upgradeable and the OSI, and the KiM, and of course, the SYM, etc. etc.

The only fault I find with the 6516 is the assembly language mnemonics chosen by Synertek. They should have followed the example laid down by Motorola and used mnemonics which specify the action, leaving the decision of where the data is coming from to the operand field.

I am currently writing a version of LISA (an interactive 6502 assembler for the APPLE II) for the 6516. I will maintain Synertek's syntax, however I will add several extensions to the syntax and in-

struction set to allow a much more regular syntax. This should prove to be a

little more pleasant to the die-hard computer scientist.

The New Instructions

LDS	M->S	(LOAD STACK POINTER FROM MEMORY)
LHA	M->AH	(LOAD HIGH ORDER ACC FROM MEMORY)
LHX	M->XH	(LOAD HIGH ORDER X-REG FROM MEMORY)
LHY	M->YH	(LOAD HIGH ORDER Y-REG FROM MEMORY)
LAX	M(X)->A	(LOAD ACC INDIRECT THROUGH X REG)
LAY	M(Y)->A	(LOAD ACC INDIRECT THROUGH Y REG)
SAY	A->M(Y)	(STORE ACC INDIRECT THROUGH Y REG)
ADD	A+M->A	(ADD W/O CARRY)
SUB	A-M->A	(SUBTRACT W/O CARRY)
AXA	A+X->A	(ADD X REG TO ACC)
AYA	A+Y->A	(ADD Y REG TO ACC)
AAX	A+X->X	(ADD ACC TO X REG)
AAZ	A+Y->Y	(ADD ACC TO Y REG)
AMX	X+M->X	(ADD MEMORY TO X REG)
AMY	Y+M->Y	(ADD MEMORY TO Y REG)
NEG	NEG(A)->A	(2'S COMPLIMENT ACC)
RLT		(ROTATE LEFT ACC)
RRT		(ROTATE RIGHT ACC)
ASR		(ARITHMETIC SHIFT RIGHT ACC)
RHL		(ROTATE AH LEFT THROUGH CARRY)
RHR		(ROTATE AH RIGHT THROUGH CARRY)
RXL		(ROTATE X REG LEFT THROUGH CARRY)
RXR		(ROTATE X REG RIGHT THROUGH CARRY)
RYL		(ROTATE Y REG LEFT THROUGH CARRY)
RYR		(ROTATE Y REG RIGHT THROUGH CARRY)
TZA	Z->AL	(TRANSFER Z TO ACC LOW)
YFC	Y->PC	(TRANSFER Y REG TO PC)
PCY	PC->Y	(TRANSFER PC TO Y REG)
XHA	AL(-)>AH	(EXCHANGE ACC BYTES)
XHY	YL(-)>YH	(EXCHANGE Y REG BYTES)
XHX	XL(-)>XH	(EXCHANGE X REG BYTES)
XXY	X(-)>Y Qx(-)>Qy	(EXCHANGE X WITH Y REGISTER)
SEF	1->F	(SET USER DEFINABLE FLAG)
CLF	0->F	(CLEAR USER DEFINABLE FLAG)
LDQ	M->Q	(LOAD Q REGISTER FROM MEMORY)
SEV	1->V	(SET OVERFLOW FLAG)
BFS		(BRANCH IF FLAG SET)
BFC		(BRANCH IF FLAG CLEAR)
JNE		(JUMP IF NOT EQUAL TO ZERO 16-BIT RELATIVE)
JEQ		(JUMP IF EQUAL TO ZERO 16-BIT RELATIVE)
PHD	A->(S)	(16-BIT ACC PUSH)
PLD	(S)->A	(16-BIT ACC PULL)
PHX	X->(S)	(16-BIT X REG PUSH)
PLX	(S)->X	(16-BIT X REG PULL)
PHY	Y->(S)	(16-BIT Y REG PUSH)
PLY	(S)->Y	(16-BIT Y REG PULL)
PHZ	Z->(S), Q->(S)	(PUSH Z REG ONTO STACK, PUSH Q REG ONTO STACK)
PLZ	(S)->Z (S)->Z	(PULL Q FROM STACK, PULL Z FROM STACK)
PHR		(COMBINATION OF PHD, PHX, PHY, AND PHZ)
PLR		(COMBINATION OF PLD, PLX, PLY, AND PLZ)
BR1		(PERFORMS A JSR (\$FFF0))
BR2		(PERFORMS A JSR (\$FFF2))
BR3		(PERFORMS A JSR (\$FFF4))
BR4		(PERFORMS A JSR (\$FFF6))
BR5		(PERFORMS A JSR (\$FFF8))

GREAT PET SOFTWARE



"Precise, humanized, well documented an excellent value" are the applauds now being given to United Software's line of software. These are sophisticated programs designed to meet the most stringent needs of individuals and business professionals. Every package is fully documented and includes easy to understand operator instructions.

DATABASE MANAGEMENT SYSTEM - A comprehensive, interactive system like those run on mainframes! Six modules comprising 42K of programming allow you to: create, edit, delete, display, print, sort, merge, etc., etc. - databases of up to 10,000 records. Printer routines automatically generate reports and labels on demand. 60 pages of concise documentation are included. Requirements - 16-32K PET and 2040 Dual Disk (printer optional). . . .Cost \$125

ACCOUNTS RECEIVABLE/PAYABLE - A complete, yet simple to use accounting system designed with the small businessman in mind. The United Software system generates and tracks purchase orders and invoices all the way through posting "controlled" accounts payable and accounts receivable subsystems. Keyed Random Access file methods makes data access almost instantaneous. The low-cost solution for the first time computer user with up to 500 active accounts. Requirements - 32K PET, Dual Disk, any 80-column printer. . . .Cost \$175

CASH RECEIPTS & DISBURSEMENTS - Makes it a breeze to track all outgoing payments made by any type of business operation. Checks are tracked by number and categorized by type of expense. Sorting, summary, and audit trails make it easy to post to general ledger. This system also categorizes incoming receipts. Uses KRAM file access method. Requirements - 32K PET, Dual Disk (printer optional). . . .Cost \$99.95

KRAM - Keyed Random Access Method - The new, ultra-fast access method for the PET Disk, provides keyed retrieval/storage of data, in either direct or sequential mode, by either full or partial key values. Written by United Software in 6502 machine code, and designed with the PET in mind, it exploits all the benefits of the PET Disk, allowing full optimization of your system. Eliminates the need for "Sort" routines! KRAM provides flexibility never seen on a micro before. KRAM is modeled after a very powerful access method used on large-scale IBM Virtual Storage mainframes. So "KRAM" all you can into your PET - it will love you for it. . . .Cost \$79.95

(Sublicenses available to software houses.)

PROGRAMS FOR ENTERTAINMENT

Space Intruders	
("Best Game of 1979")	\$19.95
Jury/Holgate	12.50
Kentucky Derby/Roulette	9.95
Alien I.Q./Tank	9.95
Tunnelvision/Maze Chase	14.95
Submarine Attack	9.95
Battle of Midway	7.95
Laser Tank Battle	9.95
Swarm	14.95

Super Startrek	14.95
PET Music Box	29.95

UNITED SOFTWARE PROGRAMS FOR BUSINESS

Checkbook	\$15.95
Mortgage	15.95
Finance	12.95
Bonds	12.95
Stock Analyzer	22.95
Stock Options	24.95
6502 Macro Assembler	49.95

Look for the RED-WHITE-BLUE United Software Display at your local computer dealer, or send check or moneyorder, plus \$1.00 shipping to:

UNITED SOFTWARE OF AMERICA

750 Third Ave.
New York, N.Y. 10017

Dealer inquiries invited

DAKIN5 UTILITIES

DAKIN5 PROGRAMMING AIDS MENU

TODAY'S DATE 09/18/79

1. THE LISTER
2. THE PEEPER
3. THE CRUNCHER
4. THE TEXT FILE COPY
5. THE PROMPTER
6. THE CALCULATOR
7. THE DISKETTE COPY
8. CHANGE TODAY'S DATE

ENTER YOUR SELECTION ->

DAKIN5 PROGRAMMING AIDS 11 MENU

TODAY'S DATE 12/17/79

1. THE SCREEN PRINTER
2. THE ARRAY EDITOR
3. THE COPIER
4. THE PATCHER
5. THE LINE CROSS REFERENCE
6. THE VARIABLE CROSS REFERENCE
7. THE CALCULATOR II
8. CHANGE TODAY'S DATE

ENTER YOUR SELECTION ->

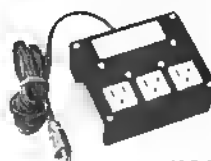
The **Cruncher** removes REM statements and compresses code in Applesoft programs. The **Prompter** is a powerful data entry subroutine that can handle both string and numeric data. Options include using commas, decimal points, and leading zeros with right-justified numerics. A maximum field length can be specified to prevent overflow in both numeric and alphanumeric fields. The **Diskette Copy** formats an output disk, copies each track, and verifies that the output matches the input. PLUS FOUR MORE UTILITIES TO AID YOUR OWN PROGRAMMING. Suggested Retail Price for **Dakin5 Programming Aids** is \$39.95.

The **Copier** copies absolutely any kind of file or program from one diskette to another. The **Variable Cross Reference** produces a cross-reference for all variable names used in an Applesoft BASIC program. The **Line Cross Reference** creates a cross-reference for an Applesoft BASIC program, showing where a given line is referenced by GOTO, GOSUB, THEN, or LIST statements. The **Patcher** allows you to display any sector of a diskette, and then to update any data within that sector. PLUS THREE MORE UTILITIES TO FACILITATE YOUR OWN PROGRAMMING. Suggested Retail Price for **Dakin5 Programming Aids 11** is \$49.95.

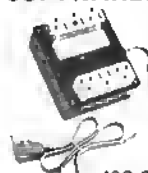
Each utility package includes a program diskette and very complete documentation. The hardware requirements are an Apple II, 48K of memory, 2 Disk II's, and a printer. Languages are Applesoft/Assembler.

See your Apple dealer or contact Dakin5 Corporation, P.O. Box 21187, Denver, Colorado 80221. Telephone: (303) 426-6090. Dakin5 developed the business application software **The Controller** for Apple Computer Inc.

DISK DRIVE WOES? PRINTER INTERACTION? MEMORY LOSS? ERRATIC OPERATION? DON'T BLAME THE SOFTWARE!



ISO-1



ISO-2

Power Line Spikes, Surges & Hash could be the culprit! Floppies, printers, memory & processor often interact! Our unique ISOLATORS eliminate equipment interaction AND curb damaging Power Line Spikes, Surges and Hash.

***ISOLATOR (ISO-1A) 3 filter isolated 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximum load, 1 KW load any socket \$54.95**

***ISOLATOR (ISO-2) 2 filter isolated 3-prong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank \$54.95**

***SUPER ISOLATOR (ISO-3), similar to ISO-1A except double filtering & Suppression \$79.95**

***ISOLATOR (ISO-4), similar to ISO-1A except unit has 6 individually filtered sockets \$93.95**

***ISOLATOR (ISO-5), similar to ISO-2 except unit has 3 socket banks, 9 sockets total \$76.95**

***CIRCUIT BREAKER, any model (add-CB) Add \$ 6.00**

***CKT BRKR/SWITCH/PILOT any model (-CBS) Add \$11.00**



PHONE ORDERS 1-617-655-1532

Electronic Specialists, Inc.

171 South Main Street, Natick, Mass. 01760



Dept. m1

Announcing...

OPTIMIZED SYSTEMS SOFTWARE

UPGRADE YOUR APPLE II® WITH A NEW SYSTEMS SOFTWARE PACKAGE

- Unified Operating System
- Disk File Manager
- Commercial Basic
- Editor/Assembler/Debugger
- Data Base Manager

Optimized Systems Software does not use Apple DOS®. OSS is a unified and complete systems software package with its own Operating System and File Manager. The Operating System, the File Manager and the Basic combined use only slightly more RAM than Apple DOS® alone. Requires 48K Apple II® with Disk II.

Operating System

- Byte and Block I/O
- Simple User Interface
- Simple Device Interface
(create your own)

Basic

- Nine Digit Precision DECIMAL Floating Point
- 32K Byte Strings
- Variable Names to 256 significant characters
- I/O Interface Statements
(no PRINT "control-D...")

File Manager

- Open, Read, Write, Delete, Lock, etc.
- Random Access via Note & Point
- File Names of Primary.Ext type

Editor/Assembler/Debugger

- Line Editor
(Edits Basic programs, too)
- Mini Assembler
- Maxi Assembler
- Disassembler
- Step, Trace, etc.

Available NOW at Special Introductory Prices

- Operating System + File Manager + Basic
- Operating System + File Manager + ASM
- Operating System + File Manager + Basic + ASM
- Operating System + Data Base Manager

Before
April 15

Beginning
April 15

\$49.95
\$49.95
\$89.95
(2nd Q)

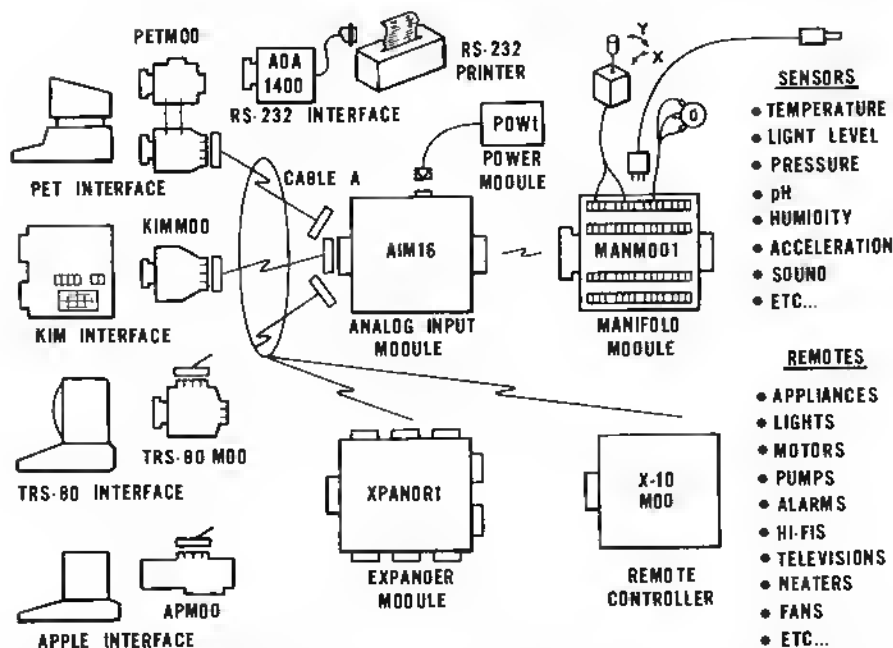
\$69.95
\$69.95
\$109.95

Order today. Add \$2.00 for shipping & handling. California residents add 6% sales tax. Visa/Mastercharge welcome. Personal checks require 2 weeks to clear.

Note: Apple II®, Apple DOS® are trademarks of Apple Computer, Inc.

Optimized Systems Software
Shepardson Microsystems, Inc.
20395 Pacifica Dr., Suite 108
Cupertino, CA 95014
(408) 257-9900

MICROCOMPUTER MEASUREMENT and



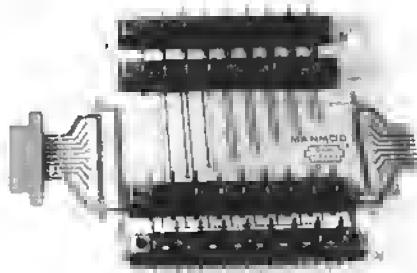
The world we live in is full of variables we want to measure. These include weight, temperature, pressure, humidity, speed and fluid level. These variables are continuous and their values may be represented by a voltage. This voltage is the analog of the physical variable. A device which converts a physical, mechanical or chemical quantity to a voltage is called a sensor.

Computers do not understand voltages: They understand bits. Bits are digital signals. A device which converts voltages to bits is an analog-to-digital converter.

Our AIM 16 (Analog Input Module) is a 16 Input analog-to-digital converter.

The goal of Connecticut microComputer in designing the uMAC SYSTEMS is to produce easy to use, low cost data acquisition and control modules for small computers. These acquisition and control modules will include digital input sensing (e.g. switches), analog input sensing (e.g. temperature, humidity), digital output control (e.g. lamps, motors, alarms), and analog output control (e.g. X-Y plotters, or oscilloscopes).

Connectors



The AIM 16 requires connections to its input port (analog inputs) and its output port (computer interface). The ICON (Input CONnector) is a 20 pin, solder eyelet, edge connector for connecting inputs to each of the AIM16's 16 channels. The OCON (Output CONnector) is a 20 pin, solder eyelet edge connector for connecting the computer's input and output ports to the AIM16.

The MANMOD1 (MANifold MODULE) replaces the ICON. It has screw terminals and barrier strips for all 16 inputs for connecting pots, joysticks, voltage sources, etc.

CABLE A24 (24 inch interconnect cable) has an interface connector on one end and an OCON equivalent on the other. This cable provides connections between the uMACSYSTEMS computer interfaces and the AIM 16 or XPANDR1 and between the XPANDR1 and up to eight AIM 16s.

XPANDR1

The XPANDR1 allows up to eight Input/Output modules to be connected to a computer at one time. The XPANDR1 is connected to the computer in place of the AIM16. Up to eight AIM16 modules are then connected to each of the eight ports provided using a CABLE A24 for each module. Power for the XPANDR1 is derived from the AIM16 connected to the first port.

Analog Input Module



The AIM 16 is a 16 channel analog to digital converter designed to work with most microcomputers. The AIM16 is connected to the host computer through the computer's 8 bit input port and 8 bit output port, or through one of the uMAC SYSTEMS special interfaces.

The input voltage range is 0 to 5.12 volts. The input voltage is converted to a count between 0 and 255 (00 and FF hex). Resolution is 20 millivolts per count. Accuracy is 0.5% \pm 1 bit. Conversion time is less than 100 microseconds per channel. All 16 channels can be scanned in less than 1.5 milliseconds.

Power requirements are 12 volts DC at 60 ma.

The POW1 is the power module for the AIM16. One POW1 supplies enough power for one AIM16, one MANMOD1, sixteen sensors, one XPANDR1 and one computer interface. The POW1 comes in an American version (POW1a) for 110 VAC and in a European version (POW1e) for 230 VAC.

TEMPSENS



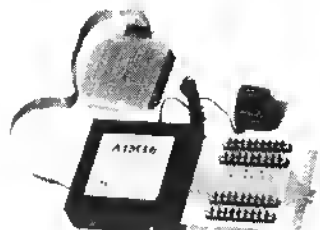
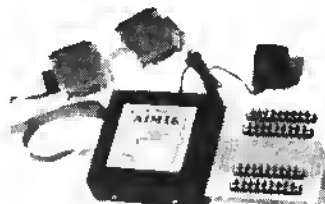
This module provides two temperature probes for use by the AIM16. This module should be used with the MANMOD1 for ease of hookup. The MANMOD1 will support up to 16 probes (eight TEMPSENS modules).

Resolution for each probe is 1°F.

CONTROL for PET, Apple, KIM, and AIM



Computer Interfaces and Sets



For your convenience the AIM16 comes as part of a number of sets. The minimum configuration for a usable system is the AIM16, one POW1, one ICON and one OCON. The AIM16 Starter Set 2 includes a MANMOD1 in place of the ICON. Both of these sets require that you have a hardware knowledge of your computer and of computer interfacing.

For simple plug compatible systems we also offer computer interfaces and sets for several home computers.

INTRODUCING SUPER X-10 MODULE

Open a door or window and turn on a light, tape recorder, alarm!

Control lab equipment. **CLOSE THE LOOP** on the real world.

AN INEXPENSIVE CONTROL SOLUTION FOR

HOME SECURITY
ENERGY CONSERVATION
GREENHOUSES
ENVIRONMENTAL CONTROL
INDUSTRIAL CONTROL
LABORATORIES

SUPER X-10 MOD SPECS

1. Remote controller
Controls up to 256 different remote devices by sending signals over the house wiring to remote modules. Uses BSR remote modules available all over the USA (Sears, Radio Shack, etc.). Does not require BSR control module. Does not use sonic link.
2. Clock/calendar
Time of day · hours, minutes, seconds
Date · month, day · automatically corrects for 28,29,30 and 31 day months.
Day of the week.
3. Digital Inputs/outputs
8 inputs · TTL levels or switch closures.
Can be used as triggers for stored sequences.
8 outputs · TTL levels



**PLUS:
CLOCK, CALENDAR,
REMOTE SEQUENCE TRIGGERS**

4. Computer interfaces
S-100: Requires one 8-bit input port and one 8-bit output port.
Requires cable assembly.
PET, APPLE, TRS-80, KIM, SYM, AIM65: Plug-in sets available - no cable assembly required.
Other: same as S-100
5. Self-contained module in metal case with its own power supply. Physical size approximately 5X6X2.

Price (until April 30, 1980): \$199.00 (S-100),
\$249.00 (other)

All prices and specifications subject to change without notice. Our 30-day money back guarantee applies.

AIM16 (16 channel-8 bit Analog Input Module)	179.00
POW1a (POWer module-110 VAC)	14.95
POW1e (POWer module-230 VAC)	24.95
ICON (Input CONnector)	9.95
OCN (Output CONnector)	9.95
MANMOD1 (MANifold MODULE)	59.95
CABLE A24 (24 inch interconnect cable)	19.95
XPANDR1 (allows up to 8 Input or Output modules to be connected to a computer at one time)	59.95
TEMPSENS2P1 (two temperature probes, -10°F to 160°F)	49.95
LIGHTSENS1P1 (light level probe)	59.95

The following sets include one AIM16, one POW1, one OCON and one ICON.

AIM16 Starter Set 1a (110 VAC)	189.00
AIM16 Starter Set 1e (230 VAC)	199.00

The following sets include one AIM16.

AIM16 Starter Set 2a (110 VAC)	239.00
AIM18 Starter Set 2e (230 VAC)	249.00

The following modules plug into their respective computers and, when used with a CABLE A24, eliminate the need for custom wiring of the computer interface.

PETMOD (Commodore PET)	49.95
KIMMOD (KIM, SYM, AIM65)	39.95
APMOD (APPLE II)	59.95
TRS-80 MOD (Radio Shack TRS-80)	59.95



Order Form

CONNECTICUT microCOMPUTER, Inc.
150 POCONO ROAD
BROOKFIELD, CONNECTICUT 06804
TEL: (203) 775-9659 TWX: 710-456-0052

The following sets include one AIM18, one POW1, one MANMOD1, one CABLE A24 and one computer interface module PETSET1a (Commodore PET).

PETSET1a (Commodore PET 110 VAC)	295.00
PETSET1e (Commodore PET 230 VAC)	305.00
KIMSET1a (KIM, SYM, AIM65 110 VAC)	285.00
KIMSET1e (KIM, SYM, AIM65 230 VAC)	295.00
APSET1a (APPLE II - 110 VAC)	295.00
APSET1e (APPLE II - 230 VAC)	305.00
TRS-80 SET1a (Radio Shack TRS-80 110 VAC)	295.00
TRS-80 SET1e (Radio Shack TRS-80 230 VAC)	305.00

[illegible]

SUBTOTAL

Handling and shipping — add per order	\$3 00
---------------------------------------	--------

Foreign orders add 10% for AIR postage	
add 7% sales tax	

TOTAL ENCLOSED	
----------------	--

NAME _____

COMPANY _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

VISA ☐ M/C ☐ Expiration date _____

Card number: _____

Card number: _____



MICRO SUBSCRIBERS SPECIAL

THANK YOU, COMMODORE...

Just about a year ago, Skyles Electric Works introduced a 15 inch wide keyboard with all PET functions on nice, plump, full-sized keytops with torsion spring action. S.E.W. knew that most owners of the PET 2001-8 couldn't remain happy with the undersized keytops and unsatisfactory spring action.

Now you have announced that, effective in January of 1980, all PETs and CBMs would feature full sized keyboards.

So, thank you, Commodore, for confirming that S.E.W. was right all along.

Meanwhile, S.E.W. hasn't been resting...and now offers with the **Big KeyBoard** an 18 inch ribbon cable at no extra charge...and a cassette tape containing BASIC and machine language programs that convert the **Big KeyBoard** to ASCII or typewriter operation, including lower case characters, and upper case through the shift key. Even a shift lock, quotations in the proper place, numbers across the top row.



*All keytops double shot for lifetime durability. Switching action uses gold cross-point contacts; torsion springs are gold plated. Housing is black heavy gauge aluminum. Entire unit can be rinsed to clean under water and left to dry. 120 day warranty. Only \$125.00**

MICRO SUBSCRIBERS SPECIAL TILL MAY 15, 1980 \$99.50*

Wouldn't You Love Your PET 2001-8 Even More With the Skyles Big KeyBoard?

Enclosed is my MICRO address label showing subscription number, please send ___Big Keyboard(s). Enclosed is a check \$_____. Charge my VISA, Mastercharge (circle one) #_____ Expires_____

*California residents: please add 6% or 6.5% sales tax as required



Skyles Electric Works

231 E. South Whisman Road
Mountain View, CA 94041
(408) 735-7891

PET Keysort

One of the most useful operations to perform on a real data base is a keysort. On the PET, due to some problems in the 'garbage collection' procedures, sorting string arrays can become very time consuming. A complete, general purpose keysorting program is presented which has many useful features and is efficient.

Rev. James Strasma
120 West King Street
Decatur, IL 62521

One of the most needed features of any business database program is a good sort routine. On the PET computer, there is also a real need for a way to sort string arrays without changing the strings. This is due to a quirk in the PET's "garbage collection" routine. PET was designed so every time a string is changed, a new string is created. Old versions are erased only after memory is filled. Then it deletes all the unneeded strings at once. As string space increases, collection time increases dramatically. With 24K of strings in memory, it can take several minutes.

Until future ROM's speed this process, it is best to avoid unneeded string manipulations. This makes a different sort program essential. For example, in an attendance program I developed, a heapsort is used. The heapsort itself takes about 20 minutes to sort 500 records. However, garbage collection adds another 2 hours! Clearly this is unacceptable.

One solution would be to define another array of integer string pointers, and sort that array. This would avoid moving strings entirely. As it happens, BASIC already stores its strings that way. Each string array is a table of pointers to another array of the actual strings. The pointers are above the program in memory, at the end of the variables. The strings are usually at the top of memory, though they may be anywhere.

I wrote a 'pointer sort' using the pointer table. It worked, but took too much memory, and had to be part of each program using it. I decided to put it in machine language instead. In the final form, it uses just under 1K of memory, at the top of memory. It resets BASIC's top of memory pointer to protect itself from

BASIC, and saves a copy of PET's zero base to protect basic from the program. The other main features of KEYSORT, are as follows:

1. extreme speed
2. simple operation
3. has defaults for all options
4. works with BASIC arrays
5. Remains until PET is reset
6. accepts any number of fields within a string
7. sorts any specified string array in memory
8. accepts any character as a field marker
9. both strings and fields may individually vary in length
10. extensive error checking

The two BASIC demonstration programs will illustrate these features. Listing #1 creates an array of random strings to sort. It does 3000 names in 28 seconds. Once you create an array to sort, merely enter 'sys(31841)' to sort it, either directly or from a program. Later, when you are ready to sort on an array other than the first in memory, try out listing #2. It uses all of KEYSORT's options at once. First, it selects the 'a\$' array as the one to sort, ignoring all other arrays. Second, it selects the '!' character as the marker between fields. Using a marker allows one string to hold about 128 separate fields at once. The array may be instantly resorted on any of these fields, as shown in sample run #2, which sorts on field #4, actually the fifth field, since there is a field #0.) You may sort by name one minute, by birthdate the next and by zip code after that.

There is no need for strings to have a fixed length. Nor is there any need for fields within strings to be any special

length. This avoids any waste of array space. KEYSORT's default field marker is the [tab] character, chr\$(9). This is easily changed, as shown in listing #2. Also there need not be any end of field marker unless you select one. Listing #1 works fine without fields. If time is very important to you. Note that using fields doubles the sort time. In return, it allows you to maintain a single data base, for several programs, and sort only the fields needed by the particular program currently in use. That saves a lot of typing time.

When you study the assembly source listing of KEYSORT, you will note a subroutine called 'spg'. This is a routine any 6502 owner can use to save up to half of zero base. By placing it at the end of the normal program flow, it only has to be called once, and its ending 'rts' then returns to BASIC.

After you assemble and save a copy of KEYSORT, call it without any arrays in memory. You will immediately see:

?array error
ready

This is KEYSORT's error message. Here it means no array was found. However, in the process, it reset Himem to protect itself from BASIC. You should do this each time you load KEYSORT, before defining strings. Otherwise they will overwrite the program. Note that if another program has already moved Himem lower than KEYSORT needs, the program leaves it alone.

If you see the '?array error' message at other times, one of several things has gone wrong. Perhaps there is no array to sort, ie. you cleared the variables. Or maybe the array has more than one

dimension—only one is allowed. (Unsorted arrays may have all the dimensions you wish.

Then again, you may have erred in poking in KEYSORT: that becomes the default for future sorts. Note that at the end of listing #2, the seven command locations are reset to zero. Unless the next sort uses the same KEYSORT features or more, you will need to zero those functions not desired in the next sort.

Both the assembly listing and the hex dump of KEYSORT here are for a 32K PET. However, the program is easily relocatable. There is no data in the body of the program, and the program does not change itself. To relocate it, merely change all of the high order bytes of 3 byte instructions, except for the one that jumps to \$C357 at \$7 01 8. This is a call to the new ROM's error message printer. Table #1 shows all the locations to change for relocation at the top of all PET model's memory. If you have an 'old ROM' PET, (8K '79 or earlier vintage), you will need to make the changes listed in table #2. You will also be limited to 256 element arrays, as the old ROM's couldn't handle more elements than that at once.

Other 6502 users with Microsoft may be able to adapt KEYSORT to their needs. My local 6502 group is converting it to the Apple, which uses a similar memory structure. It may help you to know how PET stores arrays. Each array starts with 7 housekeeping bytes. The first byte of the first array's housekeeping is addressed by 'aras' in BASIC (\$2c-2d,) low and high.) The last array ends just before the address in 'eara', (\$2e-2f). The first 2 housekeeping bytes in each array contain its name. If it is a string array, \$80 will be added to the second character of the name as a flag. Even if there is no second character, byte 2 will contain \$80. Bytes 3 and 4 are the low and high bytes of the offset from the start of the current array to the start of the next one. Byte 5 is the number of dimensions in the array, 1-3. Bytes 6 and 7 are the HIGH and low bytes respectively of the number of elements in the array. (This is backwards from the usual 6502 format.) There will be 1 more element than in the DIM statement, as the 0th element counts too. The 0th element begins immediately after the housekeeping bytes. Each element consists of 3 bytes. The first is the length of the string. The other 2 are the low and high bytes long. Also, when first dimensioned, all the length bytes and address bytes are set to zero.

I won't try to fully explain the BASIC and assembly listings of KEYSORT; they are fully commented. The only unusual feature in the BASIC programs is the use of PET's built-in 60th of a second jiffy clock, T1. When entering the assembly source, save \$3500 for the text file and

\$0200 for labels. If you have less room available, delete some comments.

If you have questions about KEYSORT, or need help, write me at the above address. Please include a stamped reply envelope. If you want a custom tape copy of KEYSORT, please send along \$5 for my time. Also, specify the starting or ending address you wish, and which ROM set you have.

Table 1: Locations to change on relocation

\$7C is found at:	\$7C62
7EFF	7F3A
\$7D is found at:	\$7C75
7CF5	7D33
7EAD	7EDC
\$7E is found at:	\$7DF7
7E48	7E87
\$7F is found at:	\$7D44
7D8F	7DA4
7DAA	7DC7
7E0C	7E68
7E9A	7EB8
	7ECB

To relocate for:

PET 4K, change 7s to 0s

PET 8K, change 7s to 1s

PET 16K, change 7s to 3s

Code will reside at Himem.

Table 2: Changes for using old ROMs Source Changes:

Line 430 ARAS .DE \$7E	Start of array space [650 & 670]
Line 440 EARA .DE \$80	End of array space [1080 & 1120]
Line 450 HIM .DE \$86	End of memory [560, 590, 610, & 630]
Line 460 ARER .DE \$85	Offset into error table [1320]
Line 470 ERRP .DE \$C359	Error msg. and stop [1330]

Object Code Changes

\$7C77 = \$7E	\$7C7B = \$7F
\$7CC7 = \$81	\$7CCF = \$80
\$7C64 = \$87	\$7C6A = \$87
\$7C6E = \$86	\$7C72 = \$86
\$7CF7 = \$85	\$7CF9 = \$59

```

100 REM> SORT DEMO #1
110 PRINT"SAMPLE RUN FOR LISTING #1":PRINT
120 S2=10:REM> ARRAY SIZE
130 DIM A$(S2)
140 REM> MAKE UP STRINGS TO SORT
150 FOR I=0 TO S2
160 A$=""
170 : FOR J=1 TO 10*RND(0)+1
180 : : A$=A$+CHR$(65+26*RND(0))
190 : NEXT
200 : A$(I)=A$
210 : PRINT I,A$
220 NEXT
230 T1=T1:REM> ZERO THE CLOCK
240 SYS(31845):REM> SORT
250 T2=T1:REM> STOP THE CLOCK
260 PRINT:PRINT"ORDER AFTER SORTING":PRINT
270 REM> PRINT THE SORTED STRINGS
280 FOR I=0 TO S2
290 : PRINT I,A$(I)
300 NEXT
310 REM> BRAG ABOUT THE TIME REQUIRED
320 PRINT:PRINT"TIME TO SORT="(T2-T1)/60"SECONDS
READY.

```

```

100 REM> KEYSORT DEMO #2
110 PRINT"SAMPLE RUN FOR LISTING #2":PRINT
120 SZ=10:REM> ARRAY SIZE
130 F1=4:REM> FIELD # TO SORT BY
140 D1=ASC(">"):REM> FIELD DELIMITER
150 S$="A$":REM> SORT ARRAY NAME
160 ZC=32731:REM> START OF Z.P. COPY
170 NMFL=ZC+2:REM> FLAGS GIVEN ARRAY
180 DFLG=ZC+3:REM> FLAGS NEW DELIM.
190 DLIM=ZC+4:REM> STORES DELIMITER
200 FDFL=ZC+5:REM> FLAGS KEY FIELD
210 FLDS=ZC+6:REM> STORES KEY FIELD #
220 DIM B$(10,2):REM> GARBAGE
230 DIM CX(10)
240 DIM D(10)
250 DIM A$(SZ):REM> ACTUAL SORT ARRAY
260 REM> MAKE UP STRINGS TO SORT
270 FOR I=0 TO SZ
280 : A$=""
290 : FOR K=1 TO 5:REM> # OF FIELDS
300 : : FOR J=1 TO 10*RNDR(0)+1
310 : : : A$=A$+CHR$(65+26*RNDR(0))
320 : : : NEXT
330 : : REM> FIELD DELIMITER
340 : : IF K<5 THEN A$=A$+CHR$(D1)
350 : : NEXT
360 : A$(I)=A$
370 : PRINT I,A$
380 NEXT
390 REM> TELL SORT FIELD # IS GIVEN
400 POKE FDFL,ASC("#")
410 REM> TELL SORT WHICH FIELD TO USE
420 POKE FLDS,F1
430 REM> GIVE SORT NEW DELIMITER
440 POKE DLIM,D1
450 REM> TELL SORT TO CHANGE DELIMITERS
460 POKE DFLG,ASC("%")
470 REM> CHANGE SORT ARRAY NAME TO BASIC
480 REM> TELL SORT SETTING NAME
490 POKE NMFL,ASC("$")
500 POKE ZC,ASC(S$):REM> CHARACTER #1
510 S2=ASC(MID$(S$,2)):REM> & #2
520 IF S2=ASC("$") THEN S2=128
530 POKE ZC+1,S2
540 T1=TI:REM> ZERO THE CLOCK
550 SYS(31841):REM> SORT
560 T2=TI:REM> STOP THE CLOCK
570 REM> CANCEL SPECIAL OPTIONS
580 FOR I=ZC TO ZC+6
590 : POKE I,0
600 NEXT
610 PRINT:PRINT"SORTED ON FIELD #"F1:PRINT
620 REM> PRINT THE SORTED STRINGS
630 FOR I=0 TO SZ
640 : PRINT I,A$(I)
650 NEXT
660 REM> BRAG ABOUT THE TIME REQUIRED
670 PRINT:PRINT"TIME TO SORT="(T2-T1)/60"SECONDS
READY.

```

Classified Ads

C1P software-Carz/Chase real-time games. \$6.95; Graphics/Billboard \$4.95; two screen clears, one for BASIC programs, one for Imm. mode \$7.95; Learning OSI Basic \$14.95. COD or Money order, SASE for catalog. Order from:
OMEGAWARE SOFTWARE
 1418 Hanson Drive.
 Normal, IL 61761
 (309) 454-1507

ZIPSOFT Presents: Number Match. A program to stretch your memory. Match sequence of numbers, colors and sounds. Different speed levels. Requires Apple II with Integer Basic and 16K. Tape \$8.95, Disk (32K req.) \$15.00. Order from:
 Z. Gittler
 50 Remsen Avenue
 Monsey, N.Y. 10952

Apple II program to output to and input from ASR 33 or 35 Teletype. Gives multiple line feeds at end of each page and waits for you to tear off roll paper or insert new single sheet. Uses game connector. Listing and instructions, \$2.00. Order from:
 Ken Ellis
 RD 8, Box 344
 York, PA 17403

Ohio Scientific. Expansion Help for C1P, C2, C4, C8 and C3. Parts, cables, boards, IC's, books, schematics. Contact:
 Silver Spur
 Electronic Communications Co.
 P.O.Box 365
 Chino, CA 91710

Attention APPLE Owners! A complete General Ledger, allowing 9 departments and 10 levels of sub-totals. Virtually unlimited capacity. Will process 1,000 postings into 70 accounts in less than 30 minutes. Contact:
 Small Business Computer Systems
 4140 Greenwood
 Lincoln, NE 68504

Brand New Commodore PET 2001-8, 32K, Video Display. Will sell at best offer; 216/543-7785.
 Dr. John W. Cook
 8670 Tanglewood Trail
 Chagrin Falls, OH 44022

Sample Run for Listing 2

Sorted on Field 4

```
0 BCHRFE>AKKYTGFC>DG>XT>NTUTHO
1 YKNJZBKT>NJVSMVI>UOF>NYCCINWGVG>YSCF
2 IHRQ>WAGQWNE>X>MP>QGCORK
3 QJ>HQMNU>UKQC>GVKPMF>VZ
4 I>GZTWM>RLH>XSBP>MLWDPWO
5 E>FL>Q>DRDT>SIVR
6 F>WO>ZNBZOHCIJ6>PTQDLIO>ZVGLAH
7 G>TFZDLKPN>NPCHNZSXNV>GSKC>BWIDSZ
8 NSDNDTKBO>LPRJWB0>VLCI>FI>OXU
9 SQPIFSBR>GKVSJCH>WDHCUQ>WQDIPC>ZSIGN
10 BLDQEND>AY>Z>L>VAOKJHR
```

```
0 G>TFZDLKPN>NPCHNZSXNV>GSKC>BWIDSZ
1 I>GZTWM>RLH>XSBP>MLWDPWO
2 BCHRFE>AKKYTGFC>DG>XT>NTUTHO
3 NSDNDTKBO>LPRJWB0>VLCI>FI>OXU
4 IHRQ>WAGQWNE>X>MP>QGCORK
5 E>FL>Q>DRDT>SIVR
6 BLDQEND>AY>Z>L>VAOKJHR
7 QJ>HQMNU>UKQC>GVKPMF>VZ
8 YKNJZBKT>NJVSMVI>UOF>NYCCINWGVG>YSCF
9 SQPIFSBR>GKVSJCH>WDHCUQ>WQDIPC>ZSIGN
10 F>WO>ZNBZOHCIJ6>PTQDLIO>ZVGLAH
```

TIME TO SORT= .083333333 SECONDS
READY.

END OF THE PASS

```
0010 ;          net keysort
0020 ;
0030 ; a multi-key sort for net basic arrays
0040 ;
0050 ;
0060 ;          by rev. james strama
0070 ;          120 w. king st.
0080 ;          decatur, il. 62521
0090 ;
0100 ;          as of feb. 14, 1980
0110 ;
0120 ;
0130 sent          .ba $7c61          ;sys(31841)
0140 ;
0150 ;
0160 ;first 5 var.s poked from basic
0170 arrnm          .de $00          ;stores array name
0180 nmfl          .de $02          ;array selected flag
0190 dfls          .de $03          ;delimiter set flag
0200 dlim          .de $04          ;delim. char.
0210 fdfl          .de $05          ;key field set flag
0220 flds          .de $06          ;sort field #
0230 ;
0240 ;
0250 ;most var.s as in basic heapsort
0260 i          .de $07
0270 j          .de $09
0280 k          .de $0b
0290 l          .de $0d
0300 m          .de $0f          ;1 by-$ lengths
0310 ln1          .de $10          ;"
0320 ln2          .de $11          ;"
0330 n          .de $12          ;elements in array
0340 r1          .de $14          ;3 by-temp. registers
0350 r2          .de $17          ;"
0360 r3          .de $1a          ;"
0370 r4          .de $1d          ;"
0380 s          .de $20          ;"
0390 v1          .de $23          ;pointer start-3
0400 ;
0410 ;
0420 ;non-dependent var.s
```

```

0430 anas      .de $2c      ;start of array space
0440 eana      .de $2e      ;end of array space
0450 him       .de $34      ;end of memory
0460 anen      .de $80      ;offset w/i error table
0470 ennp      .de $c357    ;error msg. & stop
0480 ;
0490 ;
0500 ;other labels
0510 dch       .de $09      ;tab char.
0520 locs      .de $24      ;# of locations to flip
0530 ;
0540 ;
7C61- A9 7C    0550      lda #h,sant      ;lower himem
7C63- C5 35    0560      cmp #him+1      ;unless already lower
7C65- F0 04    0570      beq hok
7C67- B0 0A    0580      bos sav
7C69- 85 35    0590      sta #him+1
7C6B- A9 61    0600 hok      lda #l,sant      ;hi, then lo
7C6D- C5 34    0610      cmp #him
7C6F- B0 02    0620      bos sav
7C71- 85 34    0630      sta #him
7C73- 20 A0 7D 0640 sav      jsr sps      ;save z.p.
7C76- A5 2C    0650      lda #anas      ;set current array ptr.
7C78- 85 15    0660      sta #r1+1      ;curr. ar. st.
7C7A- A5 2D    0670      lda #anas+1
7C7C- 85 16    0680      sta #r1+2
7C7E- A5 02    0690      lda #nmfl
7C80- C9 24    0700      cmp #'$      ;flag array name
7C82- F0 06    0710      beq okna      ;find name to match
7C84- A9 80    0720      lda #$80      ;basic's $ array flag
7C86- 85 00    0730      sta #arnm      ;use any $ array
7C88- 85 01    0740      sta #arnm+1
7C8A- A0 00    0750 okna     ldy #0
7C8C- B1 15    0760      lda (r1+1),y
7C8E- C5 00    0770      cmp #arnm
7C90- F0 06    0780      beq lok      ;1st. char. ok
7C92- A9 80    0790      lda #$80
7C94- C5 00    0800      cmp #arnm
7C96- D0 00    0810      bne wrnm      ;not right array
7C98- C8       0820 lok      iny
7C99- B1 15    0830      lda (r1+1),y
7C9B- C5 01    0840      cmp #arnm+1      ;now 2nd. char.
7C9D- 30 01    0850      bmi wrnm      ;>=$80 if $ array
7C9F- C8       0860      iny      ;>1=right
7CA0- 98       0870 wrnm     tya
7CA1- AA       0880      tax      ;name flag to x
7CA2- A5 03    0890      lda #dfls      ;flag to change delim.
7CA4- C9 25    0900      cmp #'%      ;unless=, use [tab]
7CA6- F0 04    0910      beq flc2      ;delimiter entered
7CA8- A9 09    0920      lda #dch      ;standard char.
7CAA- 85 04    0930      sta #dlim
7CAC- A5 05    0940 flc2     lda #fdfl      ;see if fields>0
7CAE- C9 23    0950      cmp #'#      ;flag set?
7CB0- F0 04    0960      beq flch      ;yes
7CB2- A9 00    0970      lda #$00      ;no, set 0 fields
7CB4- 85 06    0980      sta #flds
7CB6- A0 02    0990 flch     ldy #2      ;offset to next array
7CB8- B1 15    1000      lda (r1+1),y      ;lo

```

7CBA-	18		1010	clo		
7CB8-	65	15	1020	add *r1+1	;add to current's start	
7CBD-	85	18	1030	sta *r2+1	;to next array ptr.	
7CBF-	C9		1040	inc		
7CC0-	B1	15	1050	lda (r1+1),y	;hi	
7CC2-	65	16	1060	add *r1+2		
7CC4-	85	19	1070	sta *r2+2		
7CC6-	A5	2F	1080	lda *eana+1	;last array?	
7CC8-	C5	19	1090	cmp *r2+	;next ar. st.	
7CCA-	F0	02	1100	bex ehfd	;maybe	
7CCC-	B0	0E	1110	bos nare	;no	
7CCE-	A5	2E	1120	lda *eana	;check lo	
7CD0-	C5	18	1130	cmp *r2+1		
7CD2-	F0	02	1140	bex efnd	;end found	
7CD4-	B0	06	1150	bos nare	;not end	
7CD6-	E0	02	1160	cmp #2	;found array?	
7CD8-	B0	11	1170	bs fara	;yes	
7CDA-	90	17	1180	bcc oops	;no	
7CDC-	E0	02	1190	cmp #2	;found it?	
7CDE-	B0	0B	1200	bcs fara	;yes	
7CE0-	A5	18	1210	lda *r2+1	;no, next=current	
7CE2-	85	15	1220	sta *r1+1		
7CE4-	A5	19	1230	lda *r2+2		
7CE6-	85	16	1240	sta *r1+2		
7CE8-	18		1250	clo		
7CE9-	90	9F	1260	bcc okna	;jump	
7CEB-	A0	04	1270	ldy #4	;1 dimension allowed	
7CED-	B1	15	1280	lda (r1+1),y		
7CEF-	C9	01	1290	cmp #1		
7CF1-	F0	00	1300	bex fsiz	;ok	
7CF3-	20	A0	7D	1310	jsr sps	;restore basic
7CF5-	A2	80		1320	ldx #anar	
7CF8-	4C	57	C3	1330	jmp erre	;print error & abort
7CFB-	A0	06		1340	ldy #6	;# of elements
7CFD-	B1	15		1350	lda (r1+1),y	;lo
7CFF-	85	12		1360	sta *r	
7D01-	88			1370	dec	
7D02-	B1	15		1380	lda (r1+1),y	;hi
7D04-	85	13		1390	sta *n+1	
7D06-	18			1400	clo	;find mid element
7D07-	6A			1410	ror a	
7D08-	85	0E		1420	sta *l+1	
7D0A-	A5	12		1430	lda *r	
7D0C-	6A			1440	ror a	
7D0D-	18			1450	clo	;make % & +1
7D0E-	69	01		1460	add #1	
7D10-	85	0D		1470	sta *l	
7D12-	A5	0E		1480	lda *l+1	
7D14-	69	00		1490	add #0	
7D16-	85	0E		1500	sta *l+1	
7D18-	A5	15		1510	lda *r1+1	;current=element#0-3
7D1A-	18			1520	clo	
7D1B-	69	04		1530	add #4	
7D1D-	85	23		1540	sta *v1	
7D1F-	A5	16		1550	lda *r1+2	
7D21-	69	00		1560	add #0	
7D23-	85	24		1570	sta *v1+1	
				1580		

		1590 ;k=n		
7D25-	A5 12	1600	lda *n	
7D27-	85 0B	1610	sta *k	
7D29-	A5 13	1620	lda *n+1	
7D2B-	85 0C	1630	sta *k+1	
		1640 ;if k<1 goto l=1-1		
7D2D-	A5 0E	1650 main	lda *l+1	
7D2F-	F0 03	1660	bca ndec	
7D31-	4C B2 7D	1670 dec2	jmp dec1	
7D34-	A5 0D	1680 ndec	lda *l	
7D36-	C9 01	1690	cmp #1	
7D38-	D0 F7	1700	bne dec2	
		1710 ;r1=k		
7D3A-	A5 0B	1720	lda *k	;set k
7D3C-	85 1B	1730	sta *r3+1	
7D3E-	A5 0C	1740	lda *k+1	
7D40-	85 1C	1750	sta *r3+2	
7D42-	20 B3 7F	1760	jsr conv	;ele. # to ptr. addr.
		1770 ;s=v(k)		
7D45-	A0 00	1780	ldy #0	;r(1) to s
7D47-	B1 15	1790	lda (r1+1),y	
7D49-	85 20	1800	sta *s	
7D4B-	C8	1810	iny	
7D4C-	B1 15	1820	lda (r1+1),y	
7D4E-	85 21	1830	sta *s+1	
7D50-	C8	1840	iny	
7D51-	B1 15	1850	lda (r1+1),y	
7D53-	85 22	1860	sta *s+2	
		1870 ;v(k)=v(1)		
7D55-	A5 23	1880	lda *v1	;r(2)=v1+3
7D57-	18	1890	clc	
7D58-	69 03	1900	adc #03	
7D5A-	85 18	1910	sta *r2+1	
7D5C-	A5 24	1920	lda *v1+1	
7D5E-	69 00	1930	adc #00	
7D60-	85 19	1940	sta *r2+2	
7D62-	B1 18	1950	lda (r2+1),y	;r(1)=r(2)
7D64-	91 15	1960	sta (r1+1),y	
7D66-	88	1970	dey	
7D67-	B1 18	1980	lda (r2+1),y	
7D69-	91 15	1990	sta (r1+1),y	
7D6B-	88	2000	dey	
7D6C-	B1 18	2010	lda (r2+1),y	
7D6E-	91 15	2020	sta (r1+1),y	
		2030 ;k=k-1		
7D70-	38	2040	sec	
7D71-	A5 0B	2050	lda *k	
7D73-	E9 01	2060	sbc #1	;subtract with borrow
7D75-	85 0B	2070	sta *k	
7D77-	A5 0C	2080	lda *k+1	
7D79-	E9 00	2090	sbc #0	
7D7B-	85 0C	2100	sta *k+1	
		2110 ;if k<1 goto jeal		
7D7D-	C9 00	2120	cmp #0	
7D7F-	D0 57	2130	bne jeal	
7D81-	A5 0B	2140	lda *k	
7D83-	D0 53	2150	bne jeal	
		2160 ;r1=i		

7D85-	A5 07	2170	lda #i	;converted i to r(1)
7D87-	85 1B	2180	sta *r3+1	
7D89-	A5 08	2190	lda #i+1	
7D8B-	85 1C	2200	sta *r3+2	
7D8D-	20 B3 7F	2210	jsr conv	
		2220	;v(i)=s	
7D90-	A5 20	2230	lda #s	
7D92-	A0 00	2240	ldy #0	
7D94-	91 15	2250	sta (r1+1),y	;s to (r(1))
7D96-	08	2260	iny	
7D97-	A5 21	2270	lda #s+1	
7D99-	91 15	2280	sta (r1+1),y	
7D9B-	08	2290	iny	
7D9C-	A5 22	2300	lda #s+2	
7D9E-	91 15	2310	sta (r1+1),y	
		2320	;exchange z,n. s/r	
7DA0-	A2 24	2330	sps	
7DA2-	BD DB 7F	2340	slop	;flip z,n. locations
7DA5-	48	2350	pha	
7DA6-	B5 00	2360	lda #0,x	
7DA8-	9D DB 7F	2370	sta cpa,x	
7DAB-	68	2380	pla	
7DAC-	95 00	2390	sta #0,x	
7DAE-	0A	2400	dex	
7DAF-	10 F1	2410	bnl slop	;\$7f max
7DB1-	60	2420	rts	;end on return
		2430	;l=l-1	
7DB2-	38	2440	dec l	
7DB3-	A5 0D	2450	lda #l	
7DB5-	E9 01	2460	shc #1	; -1
7DB7-	85 0D	2470	sta #l	
7DB9-	A5 0E	2480	lda #l+1	
7DBB-	E9 00	2490	shc #0	
7DBD-	85 0E	2500	sta #l+1	
		2510	;r1=1	
7DBF-	85 1C	2520	sta *r3+2	
7DC1-	A5 0D	2530	lda #1	;conv. 1 to r(1)
7DC3-	85 1B	2540	sta *r3+1	
7DC5-	20 B3 7F	2550	jsr conv	
		2560	;s=v(1)	
7DC8-	A0 00	2570	ldy #0	; (r(1)) to s
7DCA-	B1 15	2580	lda (r1+1),y	
7DCC-	85 20	2590	sta #s	
7DCE-	08	2600	iny	
7DCF-	B1 15	2610	lda (r1+1),y	
7DD1-	85 21	2620	sta #s+1	
7DD3-	08	2630	iny	
7DD4-	B1 15	2640	lda (r1+1),y	
7DD6-	85 22	2650	sta #s+2	
		2660	;j=1	
7DD8-	A5 0D	2670	jeql	
7DDA-	85 09	2680	sta #j	
7DDC-	A5 0E	2690	lda #l+1	
7DDE-	85 0A	2700	sta #j+1	
		2710	;i=j	
7DE0-	A5 09	2720	leqj	
7DE2-	85 07	2730	sta #i	
7DE4-	A5 0A	2740	lda #j+1	

```

7DE6- 85 08      2750      sta #i+1
                  2760      ;j=j+j
7DE8- 18         2770      cld
7DE9- 26 09      2780      rol #j              ;double j
7DEB- 26 0A      2790      rol #j+1
                  2800      ;compare j & k
7DED- A5 0A      2810      lda #j+1              ;hi first
7DEF- C5 0C      2820      cmp #k+1
7DF1- F0 05      2830      beq hex
7DF3- 90 0D      2840      bcc j<k
7DF5- 4C 90 7E   2850      toj>                ;if hi=, then ok. to
7DF8- A5 09      2860      lda #j
7DFA- C5 0B      2870      cmp #k
7DFC- 90 04      2880      bcc j<k
7DFE- F0 5E      2890      beq jenk
7E00- B0 F3      2900      bos toj>
                  2910      ;if j<k then r1=j
7E02- A5 09      2920      j<k                ;j to r(1)
                  2930      lda #j
7E04- 85 1B      2940      sta #r3+1
7E06- A5 0A      2950      lda #i+1
7E08- 85 1C      2960      sta #r3+2
7E0A- 20 B3 7F   2970      jsr conv
                  2980      ;r2=v(j)
7E0D- A0 00      2990      ldy #0              ;(r(1)) to r(2)
7E0F- B1 15      3000      lda (r1+1),y
7E11- 85 17      3010      sta #r2
7E13- C8         3020      iny
7E14- B1 15      3030      lda (r1+1),y
7E16- 85 18      3040      sta #r2+1
7E18- C8         3050      iny
7E19- B1 15      3060      lda (r1+1),y
7E1B- 85 19      3070      sta #r2+2
                  3080      ;r1=v(j+1)
7E1D- 18         3090      cld
7E1E- A5 15      3100      lda #r1+1
7E20- 69 03      3110      adc #3              ;3 by. betw. ptrs.
7E22- 85 15      3120      sta #r1+1          ;w/ (r(1)) by 1 ele.
7E24- A5 16      3130      lda #r1+2
7E26- 69 00      3140      adc #0
7E28- 85 16      3150      sta #r1+2
                  3160      ;compare v(j+1) & v(j)
7E2A- A0 02      3170      ldy #2              ;copy to r(3) & r(4)
7E2C- B1 15      3180      lda (r1+1),y        ;v(j+1)
7E2E- 85 1C      3190      sta #r3+2
7E30- 88         3200      dey
7E31- B1 15      3210      lda (r1+1),y
7E33- 85 1B      3220      sta #r3+1
7E35- 88         3230      dey
7E36- B1 15      3240      lda (r1+1),y
7E38- 85 1A      3250      sta #r3
7E3A- A5 19      3260      lda #r2+2          ;v(j)
7E3C- 85 1F      3270      sta #r4+2
7E3E- A5 18      3280      lda #r2+1
7E40- 85 1E      3290      sta #r4+1
7E42- A5 17      3300      lda #r2
7E44- 85 1D      3310      sta #r4
7E46- 20 DF 7E   3320      jsr cmmr          ;compare actual $ data
                  3330      ;if v(j)>v(j+1) goto jenk

```

7E49-	A5 14	3330	lda *r1	;deciding char.s
7E4B-	C5 17	3340	cmp *r2	
7E4D-	90 0F	3350	bcc jeak	;r(2)>=r(1)
7E4F-	F0 0D	3360	bea jeak	
		3370	;j=j+1	
7E51-	18	3380	clc	;+1
7E52-	A5 09	3390	lda *j	
7E54-	69 01	3400	adc #1	
7E56-	85 09	3410	sta *j	
7E58-	A5 0A	3420	lda *j+1	
7E5A-	69 00	3430	adc #0	
7E5C-	85 0A	3440	sta *j+1	
		3450	;r1=j	
7E5E-	A5 09	3460	jeak	;j =ed k
7E60-	85 1B	3470	sta *r3+1	;conv. j to r(1)
7E62-	A5 0A	3480	lda *j+1	
7E64-	85 1C	3490	sta *r3+2	
7E66-	20 B3 7F	3500	jsr conv	
		3510	;compare v(j) & s	
7E69-	A0 02	3520	ldy #2	;r(1)) =ed v(j)
7E6B-	B1 15	3530	lda (r1+1),y	
7E6D-	85 1C	3540	sta *r3+2	;copy for s/r
7E6F-	88	3550	dey	
7E70-	B1 15	3560	lda (r1+1),y	;v(j)
7E72-	85 1B	3570	sta *r3+1	
7E74-	88	3580	dey	
7E75-	B1 15	3590	lda (r1+1),y	
7E77-	85 1A	3600	sta *r3	
7E79-	A5 22	3610	lda *s+2	;s
7E7B-	85 1F	3620	sta *r4+2	
7E7D-	A5 21	3630	lda *s+1	
7E7F-	85 1E	3640	sta *r4+1	
7E81-	A5 20	3650	lda *s	
7E83-	85 1D	3660	sta *r4	
7E85-	20 DF 7E	3670	jsr cmpr	;compare \$s
		3680	;if s<v(j) goto s<v(j)	
7E88-	A5 14	3690	lda *r1	;results here
7E8A-	C5 17	3700	cmp *r2	;r(3)'s in r(2)
7E8C-	F0 02	3710	bea j>k	;if=
7E8E-	B0 1E	3720	bos s<v(j)	;if r(1)>r(2)
		3730	;r1=1	
7E90-	A5 07	3740	j>k	;v(j)'s<=s's
7E92-	85 1B	3750	sta *r3+1	
7E94-	A5 08	3760	lda *i+1	;conv. i to r(1)
7E96-	85 1C	3770	sta *r3+2	
7E98-	20 B3 7F	3780	jsr conv	
		3790	;v(i)=s	
7E9B-	A0 00	3800	ldy #4	;s to (r(1))
7E9D-	A5 20	3810	lda *s	
7E9F-	91 15	3820	sta (r1+1),y	
7EA1-	C8	3830	iny	
7EA2-	A5 21	3840	lda *s+1	
7EA4-	91 15	3850	sta (r1+1),y	
7EA6-	C8	3860	iny	
7EA7-	A5 22	3870	lda *s+2	
7EA9-	91 15	3880	sta (r1+1),y	
7EAB-	40 2D 7D	3890	jmp main	;to top of main loop
		3900	;r2=1	

7E8E-	R5 07	3910	s<u>	lda #1	i's<u>(j)</u>'s
7E90-	85 1B	3920		sta *r3+1	
7E92-	R5 08	3930		lda #1+1	;conv. i to r(2)
7E94-	85 1C	3940		sta *r3+2	
7E96-	20 B3 7F	3950		jsr conv	
7E99-	R5 15	3960		lda #r1+1	;move to r(2)
7E9B-	85 18	3970		sta *r2+1	
7E9D-	R5 16	3980		lda *r1+2	
7E9F-	85 19	3990		sta *r2+2	
		4000	;r1=j		
7EC1-	R5 09	4010		lda #i	;conv. j to r(1)
7EC3-	85 1B	4020		sta *r3+1	
7EC5-	R5 0A	4030		lda #j+1	
7EC7-	85 1C	4040		sta *r3+2	
7EC9-	20 B3 7F	4050		jsr conv	
		4060	;v(i)=v(j)		
7ECC-	R0 00	4070		ldy #0	;j's indirect to i's
7ECE-	B1 15	4080		lda (r1+1),y	
7ED0-	91 18	4090		sta (r2+1),y	
7ED2-	C8	4100		iny	
7ED3-	B1 15	4110		lda (r1+1),y	
7ED5-	91 18	4120		sta (r2+1),y	
7ED7-	C8	4130		iny	
7ED8-	B1 15	4140		lda (r1+1),y	
7EDA-	91 18	4150		sta (r2+1),y	
7EDC-	4C E0 7D	4160		jmr leq	;back to middle
		4170	;cmpr \$s s/r		
7EDF-	R0 00	4180	cmpr	ldy #0	
7EE1-	84 0F	4190		sty *ln	
7EE3-	R6 06	4200		ldx #flds	
7EE5-	D0 0B	4210		bne notz	
7EE7-	R5 1D	4220		lda #r4	;sort on field#0
7EE9-	85 10	4230		sta *ln1	
7EEB-	R5 1A	4240		lda #r3	;1st. var. in r(3)
7EED-	85 11	4250		sta *ln2	;2nd. in r(4)
7EEF-	18	4260		clc	
7EF0-	90 70	4270		bcc fsh	;find shorter \$
7EF2-	R5 04	4280	notz	lda #dlm	;field delimiter
7EF4-	D1 1E	4290	cont	cmp (r4+1),y	;cmp flae to \$ char.
7EF6-	F0 08	4300		bex fndd	;found delim.
7EF8-	C8	4310	ont3	iny	
7EF9-	C4 1D	4320		cmp #r4	;end of \$?
7EFB-	90 F7	4330		bcc cont	;no
7EFD-	4C F3 7C	4340		jmr oops	;on error
7F00-	84 0F	4350	fndd	sty *ln	;mark current offset
7F02-	CA	4360		dex	
7F03-	F0 02	4370		bex sfld	;sort field
7F05-	B0 F1	4380		bcs ont3	;count on
7F07-	C8	4390	sfld	iny	;@ sort field
7F08-	D1 1E	4400		cmp (r4+1),y	;next char.
7F0A-	F0 05	4410		bex fnef	;field beyond sort
7F0C-	C4 1D	4420		cmp #r4	;end of \$?
7F0E-	90 F7	4430		bcc sfld	
7F10-	C8	4440		iny	
7F11-	88	4450	fnef	dex	;end on next field
7F12-	98	4460		tva	
7F13-	38	4470		sec	
7F14-	E5 0F	4480		sbc *ln	;current-start of "

7F16-	85 10	4490		sta *ln1	;len sort field
7F18-	E6 0F	4500		inc *ln	;skip delim.
7F1A-	A5 0F	4510		lda *ln	;offset from start
7F1C-	18	4520		clc	
7F1D-	65 1E	4530		adc *r4+1	;start of sort field
7F1F-	85 1E	4540		sta *r4+1	
7F21-	A5 1F	4550		lda *r4+2	
7F23-	69 00	4560		adc #\$00	
7F25-	85 1F	4570		sta *r4+2	;r(4)'s done
7F27-	A0 00	4580		ldy #0	;now other \$
7F29-	84 0F	4590		sty *ln	
7F2B-	A6 06	4600		ldx *flds	
7F2D-	A5 04	4610		lda *dlim	;only res.s differ
7F2F-	D1 18	4620	cnt2	cmp (r3+1),y	
7F31-	F0 08	4630		bex fnd2	
7F33-	C8	4640	cnt4	iny	
7F34-	C4 1A	4650		cpy *r3	
7F36-	90 F7	4660		bcc cnt2	
7F38-	4C F3 7C	4670		jmp oops	
7F3B-	84 0F	4680	fnd2	sty *ln	
7F3D-	CA	4690		dex	
7F3E-	F0 02	4700		bex sfd2	
7F40-	B0 F1	4710		bcs cnt4	
7F42-	C8	4720	sfd2	iny	
7F43-	D1 18	4730		cmp (r3+1),y	
7F45-	F0 05	4740		bex fne2	
7F47-	C4 1A	4750		cpy *r3	
7F49-	90 F7	4760		bcc sfd2	
7F4B-	C8	4770		iny	
7F4C-	88	4780	fne2	dey	
7F4D-	98	4790		tea	
7F4E-	38	4800		sec	
7F4F-	E5 0F	4810		sbc *ln	
7F51-	85 11	4820		sta *ln2	
7F53-	E6 0F	4830		inc *ln	
7F55-	A5 0F	4840		lda *ln	
7F57-	18	4850		clc	
7F58-	65 18	4860		adc *r3+1	
7F5A-	85 18	4870		sta *r3+1	
7F5C-	A5 1C	4880		lda *r3+2	
7F5E-	69 00	4890		adc #\$00	
7F60-	85 1C	4900		sta *r3+2	
7F62-	A5 10	4910	fsh	lda *ln1	;found shorter \$
7F64-	C5 11	4920		cmp *ln2	;r(4)'s in ln1
7F66-	F0 08	4930		bex ea	
7F68-	B0 0C	4940		bcs twoC	;2nd. shorter?
		4950	;which longer?		
7F6A-	85 0F	4960		sta *ln	;store least
7F6C-	A2 01	4970		ldx #1	;1st. shorter
7F6E-	D0 0C	4980		bne beas	;jump
7F70-	85 0F	4990	ea	sta *ln	
7F72-	A2 00	5000		ldx #0	;same
7F74-	F0 06	5010		bex beas	;jump
7F76-	A5 11	5020	twoC	lda *ln2	
7F78-	85 0F	5030		sta *ln	
7F7A-	A2 02	5040		ldx #2	;2nd. shorter
		5050	;init. \$ ctr.		
7F7C-	C9 00	5060	beas	cmp #0	;ok. if \$ is null

7F7E- F0 0D	5070	bea null	
7F80- A0 00	5080	ldy #0	
	5090	;compare next char.	
7F82- B1 1B	5100	nex	lda (r3+1),y
7F84- D1 1E	5110		cmp (r4+1),y
7F86- D0 24	5120	bne dif	;char.s differ?
7F88- C8	5130	iny	ino
	5140	;beyond last char.?	
7F89- C4 0F	5150	cpy #ln	
7F8B- 90 F5	5160	bcc nex	;<len
	5170	;if so,which \$ is longer?	
7F8D- E0 01	5180	null	cpy #1
7F8F- F0 09	5190		bea one<<
7F91- 10 10	5200		bpl two<<
	5210	;same	
7F93- A9 00	5220	lda #0	ino
7F95- 85 14	5230	sta *r1	;1 rts below selected
7F97- 85 17	5240	sta *r2	;from 4 options
7F99- 60	5250	rts	
	5260	;one is <	
7F9A- B1 1E	5270	one<<	lda (r4+1),y
7F9C- 85 14	5280	sta *r1	
7F9E- A9 00	5290	lda #0	
7FA0- 85 17	5300	sta *r2	
7FA2- 60	5310	rts	
	5320	;two is <	
7FA3- A9 00	5330	two<<	lda #0
7FA5- 85 14	5340	sta *r1	
7FA7- B1 1B	5350		lda (r3+1),y
7FA9- 85 17	5360	sta *r2	
7FAB- 60	5370	rts	
	5380	;found a difference	
7FAC- 85 14	5390	dif	sta *r1
7FAE- B1 1E	5400		lda (r4+1),y
7FB0- 85 17	5410		sta *r2
7FB2- 60	5420	rts	
	5430	;conversion from # to address w/i pointer array s/r	
7FB3- A5 1B	5440	conv	lda *r3+1
7FB5- 85 1E	5450		sta *r4+1
7FB7- A5 1C	5460		lda *r3+2
7FB9- 85 1F	5470		sta *r4+2
7FBB- 18	5480	clc	
7FBC- 26 1B	5490	rol *r3+1	;double it
7FBE- 26 1C	5500	rol *r3+2	
7FC0- A5 1B	5510	lda *r3+1	;+t1=t*3
7FC2- 18	5520	clc	
7FC3- 65 1E	5530	adc *r4+1	
7FC5- 85 1B	5540	sta *r3+1	
7FC7- A5 1C	5550	lda *r3+2	
7FC9- 65 1F	5560	adc *r4+2	
7FCB- 85 1C	5570	sta *r3+2	
7FCD- A5 1B	5580	lda *r3+1	;distance from an. start
7FCE- 18	5590	clc	
7FD0- 65 23	5600	adc #v1	
7FD2- 85 15	5610	sta *r1+1	;result in r(1)
7FD4- A5 1C	5620	lda *r3+2	
7FD6- 65 24	5630	adc #v1+1	
7FD8- 85 16	5640	sta *r1+2	
7FDA- 60	5650	rts	
7FDB-	5660	org	;save z.p. here
	5670	.en	

Label File

```

aras =0020
beas =7F7C
ont2 =7F2F
cont =7EF4
dch =0009
dfls =0003
eana =002E
ea =7F70
fdfl =0005
flds =0006
fne2 =7F4C
fsiz =7CFB
hok =7C6B
j =0009
jeak =7E5E
l =000D
ln2 =0011
main =7D2D
ndeo =7D34
notz =7EF2
oops =7CF3
r3 =001A
scvj =7EAE
sfd2 =7F42
spg =7DA0
two<< =7FA3

```

```

arer =0080
ckna =7C8A
ont3 =7EF8
conv =7FB3
dec2 =7D31
dif =7FAC
efnd =7C06
erro =C357
flc2 =7CAC
fnd2 =7F3B
fnef =7F11
hea =7DF8
i =0007
jck =7E02
jeal =7DD8
ln =000F
loos =0024
n =0012
nex =7F82
null =7F8D
r1 =0014
r4 =001D
sant =7C61
sfla =7F07
taj> =7DF5
v1 =0023

```

```

arnm =0000
cmpr =7EDF
ont4 =7F33
ora =7F0B
decl =7DB2
dlm =0004
ehfd =7CCE
fara =7CEB
flch =7CB6
fndd =7F00
fsh =7F62
him =0034
ieaj =7DE0
jck =7E90
k =000B
ln1 =0010
lok =7C98
nare =7C0C
nmfl =0002
one<< =7F9A
r2 =0017
s =0020
sav =7C73
slor =7DA2
two< =7F76
wrnm =7C80

```

```
//0000,8000,8000
```

1

Decision Systems



*Presenting the Other Side of the Apple II**

- ISAM-DS** An integrated set of routines for the creation and manipulation of indexed files. Retrieve records sequentially by key value or partial key value. Files never have to be reorganized.
Requires: Disk, Applesoft (32K ROM or 48K RAM)
\$50
- PBASIC-DS** A sophisticated preprocessor for structured BASIC. Gain the power of PASCAL-like logic structures at a fraction of the cost. Use all regular BASIC statements plus 14 commands and 9 new statements/structures (WHILE, UNTIL, CASE, etc.) Works with INTEGER or APPLESOFT programs.
Requires: Disk, Applesoft (32K ROM or 48K RAM)
\$35
- UTIL-DS** An Applesoft utility package that includes improved error interrupt handling (return to the statement following the one in error), a routine that selectively clears array variables (for reDIM or chaining), an internal routine that provides a 'GOSUB' facility from machine language to Applesoft and an advanced formatting routine for printing numeric values. Works with negative values; inserts commas in value, etc. Also contains a loader to put the routines into RAM with your program.
Requires: Disk, Applesoft (ROM of RAM)
\$35

(Texas residents add 5% tax)

Decision Systems
P.O. Box 13006
Denton, TX 76203

*Apple II is a registered trademark of the Apple Computer Co.

Progressive Computer Software

405 Corbin Road
York, PA 17403

PRESENTS the MACHINE LANGUAGE EXTERMINATOR

Get rid of those bugs the way the experts do. Our TEMA program allows you to easily locate errors in any machine language program.

ASCII or hex strings can be located, registers and memory locations displayed, plus many other features that are invaluable for debugging.

Available now on cassette or disk. 48K Apple required.

Cassette—\$19.95 Disk—\$24.95

APPLE ONLY

Also Available

	Cassette	Disk
One-Arm Bandit (32K)	9.95	14.95
Card Shark (INT)	7.95	9.95
High Roller (Applesoft)	7.95	9.95
Hi-Res Sub Game (32K)	14.95	19.95
Adult Games Pack	7.95	9.95
Trend Line Analysis (48K, Applesoft & ROM)	9.95	14.95

CUSTOM PROGRAMMING

We specialize in custom software. Modifications to existing programs or completely specialized to suit your needs. Write for details.

FREE! up to \$170. in merchandise with purchase of PET-CBM item !!!
FREE MERCH.

PET 16K Large Keyboard	\$ 995	\$130
PET 32K Large Keyboard	\$1295	\$170
PET 8K Large Keyboard (New)	\$ 795	\$100
PET 2040 Dual Disk (343K)	\$1295	\$170
PET 2023 Printer (pres feed)	\$ 695	\$ 70
PET 2022 Printer (trac lead)	\$ 795	\$100
KIM-1 \$159 (Add \$30 for Power Supply)		SYM-1 \$ 209.00
AXIOM EX-801 Printer-PET		\$ 477.00
2114 L 450 ns	5.35	24/4.95 100/4.45
2716 EPROM (5 Volt)		29.00
6550 RAM (for 8K Pet)		12.70
PET 4 Voice Music System (KL-4M)		34.50
All Books and Software		15% OFF
Leedex Video 100 12" Monitor		119.00

ATARI — INTRODUCTORY SPECIAL

ATARI 400, Atari 800, and all Atari Modules 20% OFF.

Heath WH-19 Terminal (fact. asm.)	770.00
Programmers Toolkit - PET ROM Utilities	44.90
PET Word Processor - Machine Language	24.00

3M "Scotch" 8" Disks	10/31.00
3M "Scotch" 5" Disks	10/31.50
Verbatim 5" Disks	10/26.50
Disk Storage Pages	10/ 3.95

Cassettes (all tapes guaranteed) Premium quality, high output low noise in 5 screw housing with labels: AGFA PE 611
C-10 10/5.95 50/25.00 100/48.00
C-30 10/7.00 50/30.00 100/57.00

Add \$1 per order for UPS shipping.
Ask for 6502, TRS-80, and S-100 Product List.

A B Computers

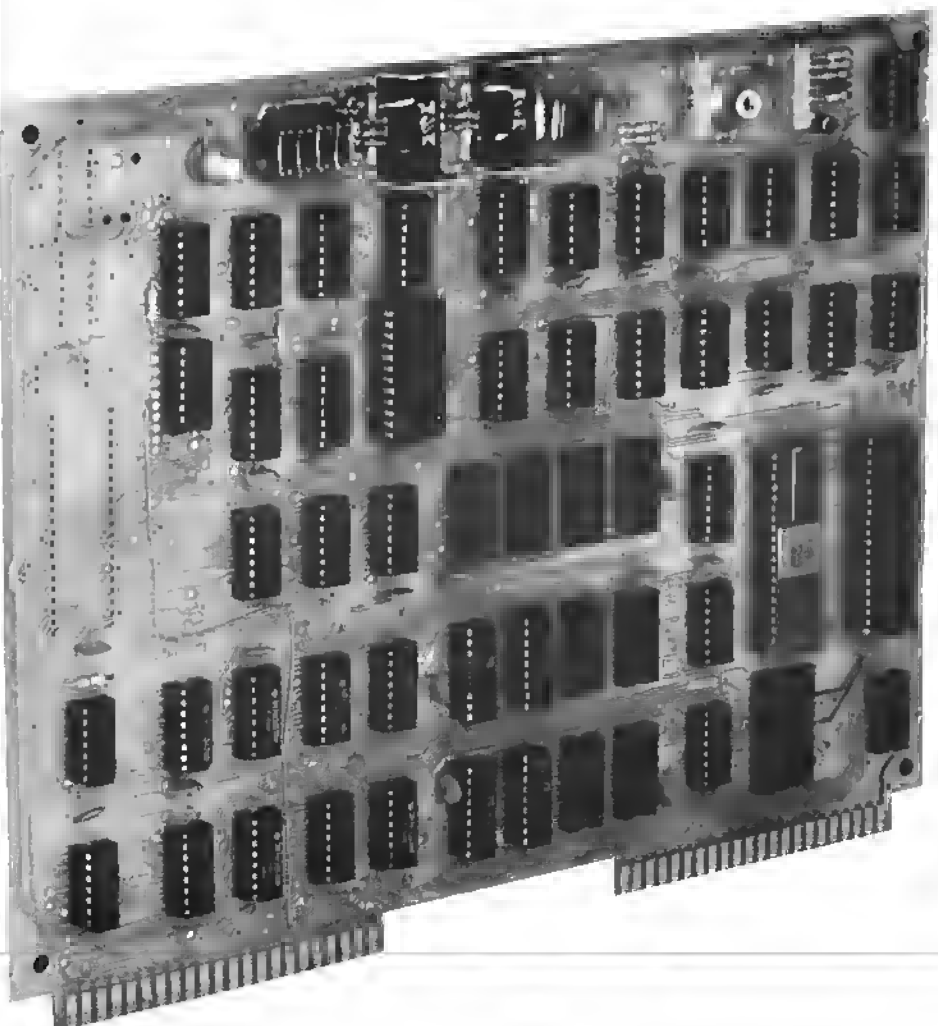
115 E. Stump Road
Montgomeryville, PA 18936
(215) 699-8386

IT'S EASY TO ADD VIDEO TO YOUR SYSTEM

VIDEO PLUSTM

VIDEO PLUS is the most versatile expansion board offered for the ASK family of micro-computers. It has many important features:

- Programmable Display Format with up to 80 characters by 24 lines.
- UPPER and lower case ASCII characters in ROM.
- Optional Programmable Character Generator with up to 128 additional user defined characters.
- Onboard Display RAM adds up to 4K bytes of RAM to your system as display memory.
- Provision for 2K EPROM memory for the ASK VIDEO PLUS Software or other support programs.
- Keyboard Interface supports an ASCII keyboard.
- Smart Video Controller supports hardware scrolling, programmable cursor, light pen, programmable video format,
- ASK VIDEO PLUS Software provides a complete control package for the AIM, SYM, and KIM. Works with the AIM/SYM Monitor, Editors, BASIC, and any program using the I/O vectors.
- Single Voltage only + 5V @ 1.5 Amps.



WE ARE LOOKING FOR A FEW GOOD COMPUTERISTS:

6502 PROGRAMMERS
TECHNICAL WRITERS
JUNIOR ENGINEERS
SENIOR TECHNICIANS

For details please call Randy at 617/256-3649 or send
your resume to:

THE COMPUTERIST
P.O. Box 3
So. Chelmsford, MA 01824

DATA CAPTURE 3.0

Is DATA CAPTURE just another smart terminal program for your Apple II[™] or Apple II Plus[™]?

NO. It is a GENIUS TERMINAL PROGRAM and is designed to be used with the Micromodem II[™].

Tired of watching data and programs scroll off the screen forever? Then DATA CAPTURE is the program for you.

- ANYTHING that appears on the screen of your Apple II can be captured. Any program or data.
- You can then save what you have captured to disk, dump it to your printer or even do simple editing with DATA CAPTURE.
- You can use DATA CAPTURE to compose text offline for later transmission to another computer. Think of the timeshare charges this will save you.
- Use DATA CAPTURE with the Dan Paymar Lower Case Adapter and you can enter lower case from the keyboard for transmission to another computer or capture both upper and lower case.
- A program is also included to convert your programs to text files for transmission using DATA CAPTURE.
- You receive two versions of the program. One is for upper case only and one for both upper and lower case use with the above adapter.

DATA CAPTURE will save you money if you are using a timesharing system because you can compose messages offline for later transmission. You can also quickly capture data for later reading, printing or editing. Requires DISK II[™], APPLESOFT II[™].

Price \$29.95

If your local dealer does not have DATA CAPTURE then order directly. We ship DATA CAPTURE within 3 working days of receipt of order and welcome your personal check. We also accept Visa and Master Charge. Add \$49.95 if you would also like to order the Dan Paymar Lower Case Adapter at the same time.

Ask for a catalog of our software.

* Apple II, Apple II Plus, Disk II and APPLESOFT II are trademarks of Apple Computer Company.

* Micromodem II is a trademark of D.C. Hayes Associates, Inc.

SOUTHEASTERN SOFTWARE

7270 Culpepper Drive
New Orleans, LA 70126

504/246-8438 504/246-7937

KIM Scorekeeper

Always on the lookout for new applications for the basic KIM-1, a general purpose, multi-player scorekeeper is presented. The techniques can be readily modified for use on a SYM-1 or AIM 65, and the scorekeeping function can be included as part of larger game programs.

Joel Swank
4655 SW 142nd, 186
Beaverton, OR 97005

Ever have a problem getting someone to keep score for your friendly game of Hearts? Well KIM would like to be a volunteer. KIM will keep up to nine separate scores for you to display and update from the keyboard. Each player can have from 0 to 9999 points, sufficient for most card games or other games needing a scorekeeper. Bridge fans can drop the low order zero from their scores (150 points for a grand slam??). I must credit the idea to a hardware project in October *Popular Electronics* by Joseph Fortuna. He used decade counters and 7-segment LED drivers to two-digit scores. A telephone dial was used to increment to counters. I immediately saw a job that KIM could do with software. Naturally with all the power of KIM available I had to improve and expand the idea.

The KIM SCOREKEEPER uses nine 2-byte memory registers to save the players' scores. Normally one of the players' scores is displayed continuously in the KIM display. The high order digit of the display is the player number, 1 to 9. The next digit is blank and the four low order digits contain that player's score. To display another player's score the PC (Player Change) key is pushed and the display goes blank. Then a number from 1 to 9 is pushed to get that player's score in the display. After a player is selected, the score can be updated. A player's score can be increased by entering the number to be added to the score and pushing the 'E' (Enter) key. Up to four digits can be entered. During entry of a number, the display shows the number being entered in the four low order digits with the two high order digits blank. Digits are shifted through the display as they are entered. If more than four digits are entered, the high order digits are shifted out and lost as in the KIM monitor.

The player's score can be decreased by pushing the 'D' (Decrease) key to set subtract mode. When the subtract mode is in effect, any number entered will be subtracted from the player's score when the 'E' key is pushed. The high order digit of the display will show a minus sign when the number being entered is to be subtracted. Subtract mode stays in effect until the '+' key is pushed to reset the program to add mode. The '+' and 'D' keys are effective anytime except when performing the player change function. If any key except 0 to 9, '+' or 'D' is entered during the update operation the display returns to the current player. The 'C' (Clear) key may be used to zero the current player's score.

As shown by the programs, SCOREKEEPER has two main display loops. One displays the current player and his score while waiting for a command from the keyboard. The other displays the number being entered while inputting digits from

the keyboard. The code is divided into subroutines for the sake of modularity and readability. The KIM subroutine GETKEY is used for communication from the keyboard, and the HEX to 7-segment conversion table in the KIM ROM is used to generate characters. The display is driven directly by the subroutine DISSEG. DISSEG is more flexible than the KIM subroutine SCANDS since it allows individual control of each segment of the KIM display. Thus any pattern can be displayed. DISSEG reads data from memory at SEGBUF and dumps it directly to the KIM display high order digit first. This subroutine could be used in a wide variety of games for KIM.

KIM SCOREKEEPER is an example of KIM's ability to replace and improve a hardware gadget. There is nothing I like more than finding a hardware function that KIM can replace with software. Someday I will calculate the weight of the hardware that my KIM has displaced.

```
0001: *****
0002: *
0003: *          KIM SCOREKEEPER          *
0004: *          VERSION 1 SEPTEMBER 1979  *
0005: *
0006: *****
0007:
0008: 0200      SCORER ORG      $0200
0009:
0010:          ZERO PAGE STORAGE
0011:
0012: 0200      PLAYER *      $0080  PLAYER SCORE TABLE
0013: 0201      MODE *      $0094  0=ADD ELSE SUBTRACT
0014: 0202      CURPLA *     $0095  INDEX TO CURRENT PLAYER
0015: 0203      CURKEY *     $0096  LAST KEY ENTERED
0016: 0204      TEMP *      $0097  REGISTER SAVE AREA
0017: 0205      INDEX *     $0098  REGISTER SAVE AREA
0018: 0206      SEGBUF *     $0099  DISPLAY BUFFER
0019: 0207      NUMBUF *     $009F  NUMBER INPUT BUFFER
0020:
0021: 0208      ZERO *      $0000
```



```

0142: 028F C9 12      CMPIM $12      PLUS KEY?
0143: 0291 C9 06      BNE CKO
0144: 0293 A9 00      LDALM ZERO
0145: 0295 B5 94      STA MODE
0146: 0297 F0 E2      BEQ UDL00P
0147: 0299 C9 0D      CKO
0148: 0299 C9 0D      CMPIM $0D
0149: 029B D0 04      BNE CKNUM
0150: 029B D0 04      STA MODE
0151: 029D B5 94      BEQ UDL00P
0152: 029F F0 0A      CKNUM
0153: 02A1 C9 0A      BCC UPLUP
0154: 02A3 90 03      RTS
0155: 02A5 60
0156: 02A5 60
0157: 02A6 A6 95      ADDUM
0158: 02AB FB
0159: 02A9 B5 B0      LDALX PLAYER GET LD BYTE OF SCORE
0160: 02A9 B5 B0      LOY MODE ADD OR SUBTRACT?
0161: 02AB A4 94      BNE SUBTRK SUBTRACT
0162: 02AD D0 0E      ADD BUFFER TO CURRENT SCORE
0163:
0164:
0165:
0166: 02AF 1B      CLC
0167: 0280 65 9F      ADC
0168: 0282 95 B0      STAZX PLAYER & SAVE
0169: 0284 EB      INX
0170: 0285 B5 B0      LDALX PLAYER GET HI BYTE
0171: 0287 65 A0      ADC
0172: 0289 95 B0      STAZX PLAYER AND SAVE
0173: 028B D8      CLO
0174: 028C 60      RTS
0175:
0176:
0177:
0178:
0179: 028D 3B      SUBTRK SEC
0180: 028E E5 9F      SBC
0181: 02C0 95 B0      STAZX PLAYER AND SAVE
0182: 02C2 EB      INX
0183: 02C3 B5 B0      LDALX PLAYER GET HI BYTE
0184: 02C5 E5 A0      SEC
0185: 02C7 95 B0      STAZX PLAYER AND SAVE
0186: 02C9 0B      CLO
0187: 02CA 60      HTS
0188:
0189:
0190:
0191:
0192:
0193: 02CB A9 C0
0194: 02CD A6 94
0195: 02CF D0 02      BNE DISMIN YES
0196: 02D1 A9 0E      LDALM ZERO
0197: 02D3 B5 99      LOXIN $02
0198: 02D5 A2 02      STX INDEX
0199: 02D7 B6 98      LDYIM $01
0200: 02D9 A0 01      NUMLUP LDAAY NUMBUF GET A BYTE
0201: 02DB B9 9F 00
0202: 02DB B9 9F 00
0203: 02DE 20 E7 02      JSR
0204: 02E1 8B      DEY
0205: 02E2 F0 F7      BEG
0206: 02E4 4C 29 03      JMP
0207:
0208:
0209:
0210:
0211:
0212:
0213:
0214:
0215: 02E7 B5 97      CVTSEC STA TEMP SAVE BYTE
0216: 02E9 4A      LSRA
0217: 02EA 4A      LSRA
0218: 02EB 4A      LSRA
0219: 02EC 4A      LSRA
0220: 02ED AA      TAX
0221: 02EE ED E7 1F      LDAAX TABLE LOAD SEGMENT CODE
0222: 02F1 A6 98      LCX INDEX GET BUFFER INDEX
0223: 02F3 95 99      STAZX SEGBUF AND STORE SEGMENT CODE
0224: 02F5 E6 98      INC INDEX NEXT BUFFER POSITION
0225: 02F7 A5 97      LDA TEMP RESTORE BYTE
0226: 02F9 29 0F      ANDIM $0F CLEAR HI NYBBLE
0227: 02FB AA      TAX
0228: 02FC ED E7 1F      LDAAX TABLE LOAD SEGMENT CODE
0229: 02FF A6 98      LDX INDEX GET BUFFER INDEX
0230: 0301 95 99      STAZX SEGBUF SAVE IN BUFFER
0231: 0303 E6 98      INC INDEX NEXT BUFFER POSITION
0232: 0305 60      RTS
0233:
0234:
0235:
0236:
0237:
0238:
0239:
0240: 0306 A5 95      DISGET LDA CURPLA GET PLAYEQR INDEX
0241: 0308 4A      LSRA
0242: 0309 AA      TAX
0243: 030A ED E7 1F      LDAAX TABLE LOAD SEGMENT CODE
0244: 030D B5 99      STA SEGBUF SAVE IN DISPLAY BUFFER
0245: 030F A9 02      LDALM $02 THIRD DIGIT
0246: 0311 B5 98      STAZX STA INDEX INIT BUFFER INDEX
0247: 0313 A4 95      LDY CURPLA PLAYER INDEX
0248: 0315 CB      INY HI BYTE
0249: 0316 B9 80 00      LDAAY PLAYER GET BYTE OF SCORE
0250: 0319 20 E7 02      JSR CVTSEC CONVERT TO SEGMENT CODES
0251: 031C BB      DEY
0252: 031D 09 80 00      LDAAY PLAYER LO BYTE
0253: 0320 20 E7 02      JSR CVTSEC CONVERT TO SEGMENTS
0254: 0323 20 29 03      JMP DISSEG DISPLAY BUFFER
0255: 0326 4C 6A 1F      JMP CETKEY GO HEAD KEYBOARD
0256:
0257:
0258:
0259:
0260:
0261:
0262:
0263:
0264:
0265:
0266:
0267:
0268:
0269:
0270:
0271:
0272:
0273:
0274:
0275:
0276:
0277:
0278:
0279:
0280:
0281:
0282:
0283:
0284:
0285:
0286:
0287:
0288:
0289:
0290:
0291:
0292:
0293:
0294:
0295:
0296:
0297:
0298:
0299:
0300:
0301:
0302:
0303:
0304:
0305:
0306:
0307:
0308:
0309:
0310:
0311:
0312:
0313:
0314:
0315:
0316:
0317:
0318:
0319:
0320:
0321:
0322:
0323:
0324:
0325:
0326:
0327:
0328:
0329:
0330:
0331:
0332:
0333:
0334:
0335:
0336:
0337:
0338:
0339:
0340:
0341:
0342:
0343:
0344:
0345:
0346:
0347:
0348:
0349:
0350:
0351:
0352:
0353:
0354:
0355:
0356:
0357:
0358:
0359:
0360:
0361:
0362:
0363:
0364:
0365:
0366:
0367:
0368:
0369:
0370:
0371:
0372:
0373:
0374:
0375:
0376:
0377:
0378:
0379:
0380:
0381:
0382:
0383:
0384:
0385:
0386:
0387:
0388:
0389:
0390:
0391:
0392:
0393:
0394:
0395:
0396:
0397:
0398:
0399:
0400:
0401:
0402:
0403:
0404:
0405:
0406:
0407:
0408:
0409:
0410:
0411:
0412:
0413:
0414:
0415:
0416:
0417:
0418:
0419:
0420:
0421:
0422:
0423:
0424:
0425:
0426:
0427:
0428:
0429:
0430:
0431:
0432:
0433:
0434:
0435:
0436:
0437:
0438:
0439:
0440:
0441:
0442:
0443:
0444:
0445:
0446:
0447:
0448:
0449:
0450:
0451:
0452:
0453:
0454:
0455:
0456:
0457:
0458:
0459:
0460:
0461:
0462:
0463:
0464:
0465:
0466:
0467:
0468:
0469:
0470:
0471:
0472:
0473:
0474:
0475:
0476:
0477:
0478:
0479:
0480:
0481:
0482:
0483:
0484:
0485:
0486:
0487:
0488:
0489:
0490:
0491:
0492:
0493:
0494:
0495:
0496:
0497:
0498:
0499:
0500:
0501:
0502:
0503:
0504:
0505:
0506:
0507:
0508:
0509:
0510:
0511:
0512:
0513:
0514:
0515:
0516:
0517:
0518:
0519:
0520:
0521:
0522:
0523:
0524:
0525:
0526:
0527:
0528:
0529:
0530:
0531:
0532:
0533:
0534:
0535:
0536:
0537:
0538:
0539:
0540:
0541:
0542:
0543:
0544:
0545:
0546:
0547:
0548:
0549:
0550:
0551:
0552:
0553:
0554:
0555:
0556:
0557:
0558:
0559:
0560:
0561:
0562:
0563:
0564:
0565:
0566:
0567:
0568:
0569:
0570:
0571:
0572:
0573:
0574:
0575:
0576:
0577:
0578:
0579:
0580:
0581:
0582:
0583:
0584:
0585:
0586:
0587:
0588:
0589:
0590:
0591:
0592:
0593:
0594:
0595:
0596:
0597:
0598:
0599:
0600:
0601:
0602:
0603:
0604:
0605:
0606:
0607:
0608:
0609:
0610:
0611:
0612:
0613:
0614:
0615:
0616:
0617:
0618:
0619:
0620:
0621:
0622:
0623:
0624:
0625:
0626:
0627:
0628:
0629:
0630:
0631:
0632:
0633:
0634:
0635:
0636:
0637:
0638:
0639:
0640:
0641:
0642:
0643:
0644:
0645:
0646:
0647:
0648:
0649:
0650:
0651:
0652:
0653:
0654:
0655:
0656:
0657:
0658:
0659:
0660:
0661:
0662:
0663:
0664:
0665:
0666:
0667:
0668:
0669:
0670:
0671:
0672:
0673:
0674:
0675:
0676:
0677:
0678:
0679:
0680:
0681:
0682:
0683:
0684:
0685:
0686:
0687:
0688:
0689:
0690:
0691:
0692:
0693:
0694:
0695:
0696:
0697:
0698:
0699:
0700:
0701:
0702:
0703:
0704:
0705:
0706:
0707:
0708:
0709:
0710:
0711:
0712:
0713:
0714:
0715:
0716:
0717:
0718:
0719:
0720:
0721:
0722:
0723:
0724:
0725:
0726:
0727:
0728:
0729:
0730:
0731:
0732:
0733:
0734:
0735:
0736:
0737:
0738:
0739:
0740:
0741:
0742:
0743:
0744:
0745:
0746:
0747:
0748:
0749:
0750:
0751:
0752:
0753:
0754:
0755:
0756:
0757:
0758:
0759:
0760:
0761:
0762:
0763:
0764:
0765:
0766:
0767:
0768:
0769:
0770:
0771:
0772:
0773:
0774:
0775:
0776:
0777:
0778:
0779:
0780:
0781:
0782:
0783:
0784:
0785:
0786:
0787:
0788:
0789:
0790:
0791:
0792:
0793:
0794:
0795:
0796:
0797:
0798:
0799:
0800:
0801:
0802:
0803:
0804:
0805:
0806:
0807:
0808:
0809:
0810:
0811:
0812:
0813:
0814:
0815:
0816:
0817:
0818:
0819:
0820:
0821:
0822:
0823:
0824:
0825:
0826:
0827:
0828:
0829:
0830:
0831:
0832:
0833:
0834:
0835:
0836:
0837:
0838:
0839:
0840:
0841:
0842:
0843:
0844:
0845:
0846:
0847:
0848:
0849:
0850:
0851:
0852:
0853:
0854:
0855:
0856:
0857:
0858:
0859:
0860:
0861:
0862:
0863:
0864:
0865:
0866:
0867:
0868:
0869:
0870:
0871:
0872:
0873:
0874:
0875:
0876:
0877:
0878:
0879:
0880:
0881:
0882:
0883:
0884:
0885:
0886:
0887:
0888:
0889:
0890:
0891:
0892:
0893:
0894:
0895:
0896:
0897:
0898:
0899:
0900:
0901:
0902:
0903:
0904:
0905:
0906:
0907:
0908:
0909:
0910:
0911:
0912:
0913:
0914:
0915:
0916:
0917:
0918:
0919:
0920:
0921:
0922:
0923:
0924:
0925:
0926:
0927:
0928:
0929:
0930:
0931:
0932:
0933:
0934:
0935:
0936:
0937:
0938:
0939:
0940:
0941:
0942:
0943:
0944:
0945:
0946:
0947:
0948:
0949:
0950:
0951:
0952:
0953:
0954:
0955:
0956:
0957:
0958:
0959:
0960:
0961:
0962:
0963:
0964:
0965:
0966:
0967:
0968:
0969:
0970:
0971:
0972:
0973:
0974:
0975:
0976:
0977:
0978:
0979:
0980:
0981:
0982:
0983:
0984:
0985:
0986:
0987:
0988:
0989:
0990:
0991:
0992:
0993:
0994:
0995:
0996:
0997:
0998:
0999:
1000:

```

```

0265: 0329 A9 7F      DISSEG LOAIM $7F      SET 6530 TO
0266: 0328 8D 41 17    STA PADD      OUTPUT
0267: 032E A0 09      LOYIM $09      SELECT DIGIT 1 FIRST
0268: 0330 A9 00      LOAIM ZERO
0269: 0332 85 98      STA INDEX      CLEAR BUFFER INDEX
0270: 0334 A6 98      LDX INDEX      GET BUFFER INDEX
0271: 0336 85 99      LDAZX SEGBUF   GET A DIGIT FROM BUFFER
0272: 0338 A2 00      LOXIM ZERO
0273: 033A 8E 40 17    STX SAD        CLEAR DISPLAY
0274: 033D 8C 42 17    STY SED        SELECT DIGIT
0275: 0340 8D 40 17    STA SAD        LITE DIGIT
0276: 0343 A2 7F      LDXIM $7F
0277: 0345 CA        WAIT DEX        LEAVE IT ON FOR A WHILE
0278: 0346 D0 FD      BNE WAIT
0279: 0348 E6 98      INC INDEX      NEXT BUFFER POSITION
0280: 034A C8        INY
0281: 034C C8        INY      SELECT NEXT DIGIT
0282: 034C C0 15      CPYIM $15      DUN YET?
0283: 034E 90 E4      SCC DISLUP     NDPE
0284: 0350 A9 00      LOAIM ZERO
0285: 0352 8D 42 17    STA SED        TURN OFF SEGB
0286: 0355 8D 41 17    STA PADD      TURN OFF 6530
0287: 0358 60      RTS
0288:
0289:
0290:
0291:
0292:
0293:
0294:
0295: 0359 0A      SHFKEY ASLA
0296: 035A 0A      ASLA          MOVE KEY TO
0297: 035B 0A      ASLA          HI NYBBLE
0298: 035C 0A      ASLA
0299: 035D A2 04      LDXIM $04      SHIFT 4 BITS
0300: 035F 2A      SHFLUP ROLA    FROM ACCUM INTO
0301: 0360 26 9F      RDL NUMBUF     NUMBER BUFFER
0302: 0362 26 A0      RDL NUMBUF     +01
0303: 0364 CA      DEX
0304: 0365 C0 F8      BNE SHFLUP
0305: 0367 60      RTS
0306:
0307:
0308:
0309:

```

Symbol Table

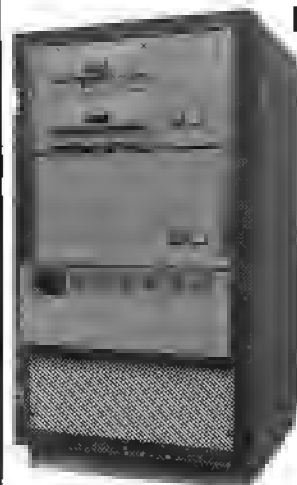
ADDUM 02A6	CKD 0299	CKNUM 02A1	CLCUR 024F
CLRLUP 0204	CURKEY 0096	CURPLA 0095	CVTSEG 02E7
DISGET 0306	DISLUP 0334	DISMIN 0203	DISNUM 02C8
DISSEG 0329	CETKEY 1F6A	GETLUP 0216	INDEX 0098
MODE 0094	NDCLR 023B	NDMNUM 0231	NDPC 0245
NOPLUS 0229	NUMBUF 009F	NUMLUP 02DB	PADD 1741
PLAYER 0080	SAD 1740	SED 1742	SCDRER 0200
SEGBUF 0099	SHFKEY 0359	SHFLUP 035F	SUBTRK 02BD
TABLE 1FE7	TEMP 0097	UDLODP 0278	UPDATE 026E
UPLUP 0278	UPPLAY 0259	WAIT 0345	ZERO 0000

April Fools On Us

We fear that a second class mail bag full of issue 21 may have been lost by the US Postal Service. If you live in the Arkansas, Louisiana, or Georgia area and did not receive your copy of MICRO, 21, please let us know.

**The Age of
Affordable
Computing
Has Arrived.**

**HAVE
YOU
?**



\$12,900

The C3-B

**OHIO SCIENTIFIC
Challenger Series**

World's largest line of microcomputers
for Personal and Business Use
There's one for every budget.
And it grows with you, too.

C1P: \$349!

**THE
ALL-IN-ONE
HOME
FINANCIER,
CALCULATOR,
TEACHER,
&
VIDEO GAME
MAKER
AT THE
WORLD'S
LOWEST PRICE!**

Ohio Scientific's
Challenger 1P is
the easy-to-use home computer.
It does a lot more for a lot less!
Just connect it to
your TV* and a
cassette
player.
That's all
there is to it. There's
nothing to build.

You can create your own personal programs.
Or select from a whole library of programs on
low-cost
cassettes.
Everything from teaching
arithmetic to spectacular
video games to balancing your
checkbook. Something for every
member of your family.

CHALLENGER 1P



*uses inexpensive TV adapter (not included)

COMPUTER SHOP

Boston
590 Comm. Ave.
(across from B.U.)
247-0700

Cambridge
288 Norfolk St.
(near M.I.T.)
661-2670

NOW YOUR APPLE II CAN PERFORM JUST LIKE THE BIG BOYS

If you're a businessman who demands ultimate performance from your Apple II, then take a look at this outstanding General Ledger Package from Small Business Computer Systems (SBCS).

It features

- 6 digit account numbers
- 31 character account name.
- Ten levels of subtotals — giving you a more detailed income statement and balance sheet.
- Departmentalizing . . . up to nine departments.
- Flexibility — adaptable to any printer and either cash or accrual accounting methods.
- Cash Journal allows a 33 character transaction description and automatically generates the appropriate offsetting entry.
- You can print the balance sheet and income statement for the current month, current quarter, or any of the previous three quarters. This year's or last year's totals are also included on the income statement. Or a special report that lists the current account balance for selected accounts.
- Higher number of entries from an external source — as many as 1,000 per session.
- No limit on entries — giving you the opportunity to make your entries as many times or as often as you want.
- With high speed printer routines and other special features of our conversion, processing performance does not decrease dramatically at the system limits.
- Look at these examples of times required to update the chart and print the audit trail.
 - With 133 item chart of accounts, 700 postings into 70 regular accounts: less than 20 min.
 - With 133 item chart of accounts, 1000 postings into 70 regular accounts: less than 30 min.
 - With 210 item chart of accounts, 1000 postings into 125 regular accounts: less than 40 min.
- Coming early this year — capability to archive up to 2,500 postings. The chart of accounts will also be archived to maintain the opening balance for the archive period.

In the final analysis, your financial statements are what this General Ledger is all about. And with this General Ledger Package you can format your own balance sheet and income statement. As well, department financial statements may be formatted differently. You have complete freedom to place titles and headings where you want them, skip lines or pages between accounts and generate subtotals and totals throughout the reports — up to ten levels if you need them.

And coming early in 1980, SBCS will present the Accounts Payable/Accounts Receivable Package you have been waiting for.

Just compare these numbers against any package on the market today:*

	5 inch disc	8 inch disc
Vendors or customers	700	1,800
Payable Transactions	350	750
Payable Invoices	380	840
Receivable Transactions	600	1,300
Receivable Invoices	600	1,300

* These are maximum numbers that you can put on a disc if you're using the disc only for these respective data files.

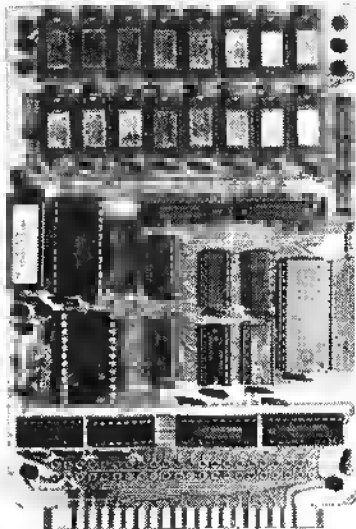
We are an authorized converter for Osborne/McGraw-Hill, providing you with business packages that will do everything the Osborne General Ledger will do in addition to many features we have added.

Call or write:

Small Business Computer Systems

4140 Greenwood
Lincoln, Nebraska 68504
(402) 467-1878

KIM/SYM/AIM-65—32K EXPANDABLE RAM DYNAMIC RAM WITH ONBOARD TRANSPARENT REFRESH THAT IS COMPATIBLE WITH KIM/SYM/AIM-65 AND OTHER 6502 BASED MICROCOMPUTERS.



ASSEMBLED/ TESTED	WITH 32K RAM	\$419.00
	WITH 16K RAM	\$349.00
	WITHOUT RAM CHIPS	\$279.00
	HARD TO GET PARTS ONLY (NO RAM CHIPS) ..	\$109.00
	BARE BOARD AND MANUAL	\$49.00

- PLUG COMPATIBLE WITH KIM/SYM/AIM-65 MAY BE CONNECTED TO PET USING ADAPTOR CABLE. 5544-E BUS EDGE CONNECTOR.
- USES +5V ONLY (SUPPLIED FROM HOST COMPUTER BUS). 4 WATTS MAXIMUM.
- BOARD ADDRESSABLE IN 4K BYTE BLOCKS WHICH CAN BE INDEPENDENTLY PLACED ON 4K BYTE BOUNDARIES ANYWHERE IN A 64K BYTE ADDRESS SPACE.
- ASSEMBLED AND TESTED BOARDS ARE GUARANTEED FOR ONE YEAR, AND PURCHASE PRICE IS FULLY REFUNDABLE IF BOARD IS RETURNED UN Damaged WITHIN 14 DAYS.
- BUS BUFFERED WITH 1 LS TTL LOAD.
- 200NSEC 4T16 RAMS.
- FULL DOCUMENTATION.

PET INTERFACE KIT \$49.00

CONNECTS THE ABOVE 32K EXPANDABLE RAM TO A 4K OR 8K PET. CONTAINS EXPANSION INTERFACE CABLE, BOARD STANDOFFS, POWER SUPPLY MODIFICATION KIT AND COMPLETE INSTRUCTIONS.

6502, 64K BYTE RAM AND CONTROLLER SET
MAKE 64K BYTE MEMORY FOR YOUR 6800 OR 6502. THIS CHIP SET INCLUDES:

- 32 M5K 4T16-3 16K X 1,200 NSEC RAMS
- 1 MC3480 MEMORY CONTROLLER
- 1 MC3242A MEMORY ADDRESS MULTIPLEXER AND COUNTER.
- DATA AND APPLICATION SHEETS. PARTS TESTED AND GUARANTEED.

\$295.00 PER SET

16K X 1 DYNAMIC RAM
THE MK4116-3 IS A 16,384 BIT HIGH SPEED NMOS DYNAMIC RAM. THEY ARE EQUIVALENT TO THE MOSTEK, TEXAS INSTRUMENTS, OR MOTOROLA 4116-3.

- 200 NSEC ACCESS TIME, 375 NSEC CYCLE TIME.
- 16 PIN TTL COMPATIBLE.
- BURNED IN AND FULLY TESTED.
- PARTS REPLACEMENT GUARANTEED FOR ONE YEAR.

\$8.50 EACH IN QUANTITIES OF 8

BETA
COMPUTER DEVICES

1230 W. COLLINS AVE.
ORANGE, CA 92668
(714) 633-7280

CALL-RESIDENTS PLEASE ADD 6% SALES TAX.
MASTERCARD & VISA ACCEPTED. PLEASE
ALLOW 14 DAYS FOR CHECKS TO CLEAR. BANK
PHONE ORDERS WELCOME.

**ALL ASSEMBLED BOARDS AND MEM-
ORY CHIPS CARRY A FULL ONE YEAR
REPLACEMENT WARRANTY.**

ACTION, STRATEGY, AND FANTASY for the **SERIOUS** games player and his **APPLE II**

Brain Games - 1 demands ingenuity.

Two players bombard radioactive material with protons and electrons until it reaches critical mass and sets up a **Nuclear Reaction**. **Dodgem** requires you to outmaneuver another player to get your pieces across the board first. **Dueling Digits** and **Parrot** challenges your ability to replicate number and letter sequences. **Tones** lets you make music with your Apple (16K) CS-4004 \$7.95. **Strategy Games** and **Brain Games** are on one disk (16K) CS-4503 \$14.95.

Strategy Games - 1 keeps games players in suspense.

You and your opponent trail around the screen at a quickening pace attempting to trap each other in your **Blockade**. A 7 category quiz game will certify you as a **Genius** (or an errant knave!). Beginners will meet their master in **Checkers**. **Skunk** and **UFO** complete this classic collection (16K) CS-4003 \$7.95

Know Yourself through these valid self-tests.

Find out how your life style effects your **Life Expectancy** or explore the effects of **Alcohol** on your behavior. **Sex Role** helps you to examine your behavior and attitudes in light of society's concept of sex roles. **Psychotherapy** compares your feelings, actions, and phobias to the population's norms and **Computer Literacy** tests your microcomputer savvy. A fun and instructional package (16K) CS-4301 \$7.95. **Know Yourself** and **CAI Programs** are on one disk (16K) CS-4503 for \$14.95

THINK



IMAGINE



You're in command in Space Games - 1.

Maneuver the TIE fighters into your blaster sights and zap them with your lasers to save the rebel base camp from annihilation in **Star Wars**. **Rocket Pilot** is an advanced real time take off and landing game. High resolution graphics, exploding saucers and sound effects add to the suspense as you repel the **Seucer Invasion**. Finally, a bonus graphics demonstration, **Dynamic Bouncer** (16K) CS-4001 \$7.95. **Space Games** and **Sports Games** are on one disk (16K) CS-4501 for \$14.95

ACTION

Sports Games - 1 puts you in the Apple World Series

Take the field in the **Great American Computer Game**. Mix up your pitches to keep the batter off balance. Move your fielders to snag the ball before he gets to first. Balls and strikes, double plays, force outs, and errors let you play with a realistic strategy. Also in the line up—**Slalom**, a championship downhill ski race, **Torpedo Alley**, and **Darts** (16K) CS-4002 \$7.95. **Spec Games** and **Sports Games** are on one disk (16K) CS-4501 for \$14.95

It's easy to order SENSATIONAL SOFTWARE for your Apple II.

Send payment plus \$1.00 shipping and handling in the U.S. (\$2.00 foreign) to Creative Computing Software, P.O. Box 789-M, Morristown, N.J. 07960. N.J. residents add \$1.00 sales tax. Visa, Master Charge and American Express orders may be called in toll free to 800-631-8112 (in N.J. 201-540-0445).

OSI BASIC in ROM

While the various Microsoft BASICs are easy to use, they are difficult to understand due to an intentional lack of documentation. To help understand your OSI BASIC, a table of the locations of the subroutines to service the main commands is presented. The program which generated the table is provided as a starting point for you to explore your BASIC.

E.D. Morris, Jr.
3200 Washington
Midland, MI 48640

A previous article in Micro 18:9 by S.R. Murphy gave a peek into OSI BASIC in ROM by listing a number of scratch pad locations in page zero. In the present article, I wish to delve further into the inner workings of BASIC by explaining the dispatch table.

At the bottom of the BASIC ROMs, between \$A000 and \$A083, is a list of addresses known as the dispatch table. These are the starting addresses of all the machine subroutines needed to carry out the BASIC keywords such as END, FOR, NEXT etc. The addresses are in hexadecimal in the normal machine format of low byte first followed by the high order byte. For example, starting at \$A000 you find the data:

\$A000	39
\$A001	A6
\$A002	55
\$A003	A5

Thus the first two entries in the dispatch table are \$A639 and \$A555. These point to subroutines in the BASIC ROMs.

Now we need to know what each subroutine does. Conveniently there is another table starting at \$A084 containing a list of all the BASIC keywords. The first entries in this table are:

\$A084	45
\$A085	4E
\$A086	C4
\$A087	46
\$A088	4F
\$A089	D2

Except for the C4 and D2, the data looks like ASCII code. If the high order bit

is removed from C4 and D2, then it is ASCII code for ENDFOR. You can demonstrate the list of keywords for yourself by running the program:

```
10 FOR X=41092 TO 41315
20 Y=PEEK(X)
30 PRINT CHR$(Y);
40 NEXT
```

If you have the OSI graphics character generator, the last letter of each word will be a graphics character instead of a letter. The high bit being set is used to separate the entries in the word list. To convert these to letters and leave a space between key words, add the following line to the above program:

```
25 IF Y > 127 THEN PRINT
   CHR$(Y-128);Y=32
```

Now we have two lists, one of addresses and one of functions. These can be combined to give an address for each function.

END	\$A639
FOR	\$A555

However things are not quite that simple. Unfortunately the two tables are not strictly in the same order. Also some of the address entries refer to the subroutine location and others to the location, less one. The address table is further complicated in the case of the arithmetic operators by a third entry which is the precedence value.

Following is a BASIC program that sorts out these quirks and outputs a list of BASIC KEYWORDS together with the hex address of the machine code

associated with that keyword. Notice that the program does not contain data statements, rather PEEK's directly at your BASIC ROM's. The program steps through the dispatch table printing out each address. The value of Q is added to each address and is either 1 or 0. The correct keyword is found by PEEKing at D until a character is found with the high bit set.

The subroutine at line 500 converts a binary word into ASCII digits for printing.

For those of you who have trouble with this program or for those who have a sore index finger from typing in that 24K game program, I am providing an output listing. However I urge you to run it yourself to prove all this stuff is really "In there." The BASIC program also contains information about the location and structure of the two tables.

Looking at the sample run, the addresses for END and FOR found earlier, are incorrect by one byte. Users of the USR function know that the subroutine address must be placed at \$000B and \$000C. The dispatch table associates location \$000A with the USR function. Location \$000A contains 4C or JMP which completes the three byte instruction.

It is interesting to note that the BASIC keyword table is identical to a numerical listing of the BASIC tokens (MICRO 15:20). The keywords TAB, TO, THEN, and STEP are missing from the dispatch table. However these commands are never used alone but always occur with another BASIC keyword (PRINT, FOR, IF and FOR-NEXT). The purists will note the absence of AND, OR, GREATER, LESS

and EQUALS. I must confess, these did not fit neatly into my BASIC program.

If you have ever tried to make sense of "that 8K block of data up there at \$A000," it looked like a hopeless task. With the dispatch table at hand, you can break it down and attack one function at a time.

Sample Run (Program output listing)

A63A	END
A556	FOR
AA40	NEXT
A70C	DATA
A923	INPUT
AD01	DIM
A94F	READ
A7B9	LET
A6B9	GOTO
A691	RUN
A73C	IF
A61A	RESTORE
A69C	GOSUB
A6E6	RETURN
A74F	REM
A638	STOP
A75F	ON
A67B	NULL
B432	WAIT
FFF4	LOAD
FFF7	SAVE
AFDE	DEF
B429	POKE
A82F	PRINT
A661	CONT
A4B5	LIST
A68C	CLEAR
A461	NEW
B7D8	SGN
B862	INT
B7F5	ABS
000A	USR
AFAD	FRE
AFCE	POS
BAAC	SOR
BBC0	RND
B5BD	LOG
BB1B	EXP
BBFC	COS
BC03	SIN
BC4C	TAN
BC99	ATN
B41E	PEEK
B38C	LEN
B08C	STR\$
B3BD	VAL
B39B	ASC
B2FC	CHR\$
B310	LEFT\$
B33C	RIGHT\$
B347	MID\$
B46F	+
B458	-
B5FE	*
B6CD	/
BAB6	↑

These subroutines are available to use if you are into machine code programming. Mr. Murphy is wrong: OSI users are not disinclined to explore their machines. The problem, until now, has been that too lit-

tle information was available. So let's dig into QSI's BASIC and publish a complete memory map similar to those already out for the PET and APPLE.

BASIC Program

```

10 Q=1:D=41092
20 FOR C=40960 TO 41060 STEP 2
25 IF C=41016 THEN Q=0:D=41237
30 X=PEEK(C+1):GOSUB 500
40 X=Q+PEEK(C):GOSUB 500
50 PRINT" ";
60 X=PEEK(D)
70 D=D+1
80 IF X<128 THEN PRINT CHR$(X);:GOTO60
90 X=X-128
100 PRINTCHR$(X)
110 NEXT C
115 D=41224
120 FOR C=41062 TO 41074 STEP 3
130 X=PEEK(C+2):GOSUB 500
140 X=1+PEEK(C+1):GOSUB 500
150 PRINT" ";
160 X=PEEK(D)
170 D=D+1
180 IF X<128 THEN PRINT CHR$(X);:GOTO 160
190 X=X-128
200 PRINT CHR$(X)
210 NEXT C
220 END

500 REM PRINT SUB
510 H=INT(X/16)
520 L=X-16*H
530 IF H<10 THEN H=H+48:GOTO 550
540 H=H+55
550 IF L<10 THEN L=L+48:GOTO 570
560 L=L+55
570 PRINT CHR$(H);CHR$(L);
580 RETURN

```

Would you like to become a Micro Dealer?

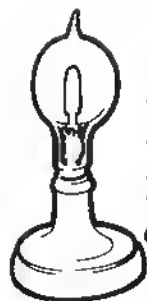
MICRO is a quality 6502 magazine. Our current dealers report that having MICRO available for sales in their stores actually helps sell 6502 based systems.

We require a minimum quantity of only 10 copies per month, and we offer a standard trade discount.

Other items available are the Best of MICRO, Volumes 1 and 2, and All of MICRO.

If you are interested in becoming a MICRO dealer, please write to:

MICRO
Box 6502
Chelmsford, MA
01824



Said the Toolkit to the Word Processor: "You're in My Space!" Said the Word Processor to the Toolkit: "Let's Share...here's

Socket 2 Me™!"

From the original producer of peripherals exclusively for PET lovers everywhere . . . the device that allows you to select between the BASIC Programmer's Toolkit and the Commodore Word Processor II while they occupy the same address space.



The **Socket 2 ME** . . . doubles your memory expansion in a single socket. It's a 2.5" x 2.75" board that fits neatly into the Toolkit/Word Processor socket on the main logic board of all *new* PETs. Then both the Word Processor and the Toolkit plug into the **Socket 2 ME**.

A miniature slide switch — part of the kit — mounts with double-stick tape (supplied) to the front part of the right side of the PET base, almost hidden by the overhang of the top of the PET cabinet. The slide switch is connected to the **Socket 2 ME** by a special cable (also supplied) . . . and you're up and running.

Up and running; installation took only a minute or so. Flip the switch from Toolkit to Word Processor. And back. No need to open the PET.

Complete with the first-rate installation and operating instructions you've come to expect from all Skyles documentation.

YOU HAVE AN ORIGINAL 2001-8 PET?

No problem. The **Socket 2 ME** interfaces with the BASIC Programmer's Toolkit model TK 160E or TK 160S connector board, the Word Processor II interfaces with the **Socket 2 ME**, the slide switch is placed on the PET base. Then, as long as the PET 2001-8 has at least 8K of memory expansion, the system is up and running.

YOU HAVE A COMPUTHINK DISK AND YOU WANT A BASIC PROGRAMMER'S TOOLKIT?

*How would you like to switch between the Computhink and the Toolkit with a single **SYS** command?*

Just add two small jumpers to the Computhink system and a short program to the DOS diskette. Plug in the BASIC Toolkit TK 80E, enter the **SYS** command and your system is up and running.

NO ROOM ON THE PET 2 COMPUTHINK DISK BOARD?

All your sockets are booked? Fret not; Skyles comes to the rescue. **Skyles Electric Works** now has a modified EPROM board available with sockets for the Toolkit and Computhink ROM chips. Plug in the ROMs, add a jumper (supplied) to the PET 2 Computhink disk board, plug the new EPROM board into the Computhink disk board. Power up and enter a short switching program into the DOS diskette. Switch between the Computhink disk and the Toolkit with a single **SYS** command.

ORDER NOW — with Skyles' 10-day money-back guarantee:

Socket 2 ME: \$22.50*
Commodore Word Processor II: \$100.00*
Commodore Word Processor III: \$200.00*

BASIC Programmer's Toolkit
Model TK 80ED \$85.00*
Model TK 160ED \$95.00*†

*Add \$2.50 to each for shipping and handling.

†Note: If Computhink EPROM board is returned, after purchase of TK 160ED, Skyles will refund \$20.00.

"Socket 2 ME" is the trademark of Skyles Electric Works.

California residents: please add 6% or 6.5% sales tax as required

VISA, MASTERCARD ORDERS CALL (800) 538-3083 (except California residents)
CALIFORNIA ORDERS PLEASE CALL (408) 257-9140



Skyles Electric Works

231 E. South Whisman Road
Mountain View, CA 94041
(408) 735-7891



Now You Can Have INPUT/OUTPUT

For Your Apple Computer

DUAL AXIS JOY-
STICK AND PUSH-
BUTTON MOD-
ULE for Games,
Graphics, and
Experimental Pro-
gram Input



MICROBOX

CONNECTS DIRECTLY
to the Apple Game
Socket. ACCEPTS 2
Dual Axis MICROSTIKS
or 4 Paddles

MICROSTIK

EXTRA LONG, HEAVY DUTY
cables and connectors

ISOLATED SWITCH-
ING of 4 AC loads or
relays from a basic
program 4 LED
status indicators.
Toggle Switch input
(sw3)

**MICROBOX AND MICRO-
STIK PROVIDE APPLE
OWNERS WITH THE
HARDWARE TO EXPLORE
THE INPUT/OUTPUT CA-
PABILITIES OF THEIR
COMPUTERS.**

A SIMPLE COMMAND from the Apple keyboard or a Basic Program can switch an external device. Connect AC loads, such as lamps, motors, relays or solenoids directly through the MICROBOX's 4 AC OUT-LETS. Loads can range from 0 to 220VAC and draw up to 200 Watts each. Solid State Switching ISOLATES the load from your Apple for complete safety. Four LEDs provide a visual on/off status of each load.

A Complete Instruction/Tutorial Manual is included with the MICROBOX.

REAL-TIME INPUT

The MICROSTIK is a sturdy, two axis joystick. Metal Cable Connectors assure trouble free usage over time, and enable extension cables to be added easily. Use the MICROSTIK to add real-time input to your game, graphic or experimental programs. Each MICROSTIK contains a PUSHBUTTON for added input possibilities.



MICROBOX and MICROSTIK sit comfortably on, or aside the Apple Computer. They have been designed to match the Apple in color and design.

**ORDER TODAY AND CON-
NECT YOUR APPLE TO
THE OUTSIDE WORLD.**

The MICROBOX and MICRO-
STIK can be purchased at
most computer stores or can
be ordered directly by mail or
through our convenient 24
hour telephone service.

TELEPHONE:
(703) 471-4291

Order the MICROSET and
receive the MICROBOX, 2
MICROSTIKS, the Manual
and Cassette, and SAVE \$25.

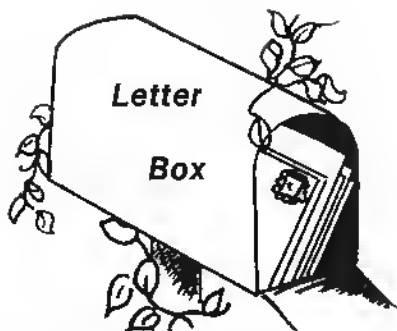
MICROBOX	\$109.95
MICROSTIK (each)	34.95
Demo Cassette	9.95
MICROSET	164.95
12 ft. Ext. Cables	6.95
Relay Modules	WRITE
Solenoid Modules	WRITE

Va. residents add 4% sales tax

MASTER CHARGE, VISA accepted
NO C.O.D.s

CJM Industries, Dept. MB
316B Victory Drive
Herndon Industr Park
Herndon, Va. 22070

CJM
INDUSTRIES



Dear MICRO magazine,

My Dad and I have had an APPLE II for about 9 months. During this time I have learned of the special joys and sorrows that only computer people can appreciate or experience. This poem was born out of long hours at the keyboard. I hope you like it and feel that it is worth publishing.

Ode to My Disk

I always see verses praising the Apple
But who sees the time saved by Disk II, it's ample?
While Apple sits waiting to digest the data
That trusty old "breadbox" spins round without
breakdown.

It hasn't been long since I've bought my Disk II
But I know it's worth it and so do you.

I tried Panasonics and Hitachis too
Resetting and loading my Apple I'd do.

Frustrating it was and my hair I did pull
So soon I did tire of ERR MEM FULL.

So now my Hitachi sits dusty and wan
And softly clicks Disk II, no ERR coming on.

Donna Marie Andert
Connelly High School
Anaheim, CA 92801

Dear Editor,

Most articles that are submitted to MICRO are claimed by their authors to execute correctly. The following program has been extensively de-bugged and is guaranteed to run *neither* on a PET *nor* on an OSI microcomputer.

```
10 FOR X=1 TO 10
20 IF X=5 THEN 40
30 NEXT X
40 REM
100 FOR Y=1 TO 10
110 FOR X=1 TO 10
120 NEXT X
130 NEXT Y
READY.
```

Can you figure out what is wrong here? If not, the answer is given in the next column

E.D. Morris, Jr
Midland, MI 48640

This program was originally part of a 200 line game program with a "small bug." Through a bit of detective work, I narrowed the bug down to these eight lines. In the original game, these lines occurred in widely different sections of the program and appeared not to be related. When the program is executed, the computer will halt indicating "NEXT WITHOUT ERROR IN LINE 130".

This message is most confusing since line 100 clearly contains a "FOR Y". The program will run if lines 100 and 130 are deleted. Something appears to be wrong with the "Y" loop. If "X" is made the outer loop and "Y" is the nested loop, the program will run without error.

This is all a wild goose chase! Nothing is wrong with the "Y" loop. The first real hint of the cause is that replacing the variable "X" in lines 110 and 120 with a different variable, say "Z", solves the problem. The real culprit is line 20 where the program jumps out of a loop before finishing it. It is simple to see here in an eight line program, but not so obvious in a large program. The problem occurs when a variable from an unclosed loop is used again in a nested loop.

The moral of the story is to close loops whenever possible. For example, line 20 could have been:

```
20 IF X=5 THEN Z=X : X=10 : GOTO 30
```

If you can't close the loop, at least avoid using that variable in another loop.

And here is another poem from a reader, sent to us in May, 1979. We hope that he remembered to renew his subscription.

End of Subscription

There once was a town, Albuquerque,
Wherein lived a genuine turkey
Who, on learning his MICRO had died,
Lost what little was left of his pride.

Hadn't realized how close was the end.
Still, he took out his pencil and penn'd
"Mr. Tripp, won't you give me a chance?
My check will disprove miscreance."

Nelson E. Ingersoll
Albuquerque, NM 87110

We at MICRO would like to thank Donna, Earl, Nelson and all of our readers for their contributions. While all of the letters that we get are not as entertaining or as fun as these, they all certainly give us some things to think about. We welcome reader input and we encourage you to write to us with your comments, and suggestions at any time. We hope to run the Letterbox column in every issue, but it all depends on what we get from you.

The MICRO Staff

Apple-Doc

By Roger Wagner

An Aid to the Development and Documentation of Applesoft Programs

This 3 program set is a must to anyone writing or using programs in Applesoft! It not only provides valuable info. on each of your programs, but allows you to change any element throughout the listing almost as easily as you would change a single line!!

With Apple-Doc you can produce a list of every variable in your program and the lines each is used on, each line called by a GOTO, GOSUB, etc., in fact, every occurrence of almost anything!

You can rename variables, change constants and referenced line #'s, or do local or global replacement editing on your listing.

In fact, we guarantee that after purchase, if you don't feel APPLE-DOC is one of the most valuable programs in your library we will even refund your money! (Upon return of product.)

Unheard of? Yes! But that's how good APPLE-DOC really is!

That's not all!! Send for free info. or visit your nearest Apple dealer.

Only \$24.95 Please specify diskette or tape.
(Calif. residents add 6% Sales Tax)

Available from your local computer store or:

Southwestern Data Systems
P.O. Box 582-M
Santee, CA 92071
(714) 562-3670

(Dealer inquiries invited)

PET Printer Adapter



CMC's ADA 1400 drives a printer with an RS-232 interface from the Commodore PET IEEE-488 bus. The ADA 1400 is addressable, works with the Commodore disk and prints upper and lower case ASCII.

A PET IEEE type port is provided for daisy-chaining other devices.

A cassette tape is included with programs for plot routines, data formatting and screen dumps. The ADA 1400 sells for \$179.00 and includes a PET IEEE cable, RS-232 cable, power supply, case, instructions and software.

Order direct or contact your local computer store.

VISA AND M/C ACCEPTED - SEND ACCOUNT NUMBER, EXPIRATION DATE AND SIGN ORDER.
ADD \$3 PER ORDER FOR SHIPPING & HANDLING - FOREIGN ORDERS ADD 10% FOR AIR POSTAGE.

CONNECTICUT microCOMPUTER, Inc.
150 POCONO ROAD
BROOKFIELD, CONNECTICUT 06804
TEL: (203) 775-9659 TWX: 710-456-0052

STONEWARE for APPLE II*

For the Serious Business or Home User: MICRO MEMO

MICRO MEMO is the first sophisticated "Desk Calendar" program to make good use of your computer's power.

- * Micro Memo includes one time, weekly, monthly, semi-annual and annual reminders.
- * Monthly reminders may be for fixed or "floating" dates (e.g. 1st Saturday of every month).
- * Each reminder allows choice of one week, 2 week or 1 month advance notice—reminds you ahead of time to prepare for meetings, purchase tickets, make reservations, etc.
- * Micro Memo includes "shorthand" for fast memo entry, greater capacity
- * Micro Memo will display or print any day's or week's reminders
- * Micro Memo is a "perpetual" calendar—automatically creates new months with all appropriate mementos (birthdays, anniversaries, monthly meetings, etc.) as past months are dropped—system holds full year's reminders on one disk
- * Micro Memo "knows" most major holidays
- * Supports Mountain Hardware clock (optional)
- * "Bomb Proof" menu driven command and data entry.
- * Requires 48K, disk, RAM or ROM Applesoft.

\$39⁹⁵

STONEWARE
Microcomputer Software
P.O. Box 7218, Berkeley, CA 94707
(415) 548-3763

And Just for Fun: TRANQUILITY BASE



TRANQUILITY BASE is a fast, high resolution Lunar Lander game by Bill Budge, creator of Apple's "Pony Arcade." TRANQUILITY BASE is just like the popular arcade game, including multiple moonscapes, craft rotation, and zoom in for a close-up view as you approach the Lunar surface.

TRANQUILITY BASE requires 32K and disk

Available at your favorite computer store or direct from STONEWARE (add \$2 shipping & handling; Calif. residents add sales tax. Visa & MasterCard accepted, no C.O.D.'s).

DEALER INQUIRIES INVITED

* Apple II is a Trademark of Apple Computer, Inc.

The MICRO Software Catalogue: XIX

Mike Rowe
P.O. Box 6502
Chelmsford, MA 01824

Name: **Dakin 5 Programming Aids II**
System: **Apple II**
Memory: **48K**
Language: **Assembler / Applesoft II**
Hardware: **Apple II, 2 Disk II's, and Printer**

Description: Set of seven programs: 1) *Copier* - copies absolutely any kind of file or program from one diskette to another. 2) *Variable Cross Reference* - creates a cross-reference for all variable names used in an Applesoft BASIC program, showing all line numbers where a given variable name is used. 3) *Line Cross Reference* - creates a cross-reference for all referenced lines in an Applesoft BASIC program, showing where a given line is referenced by GOTO, GOSUB, THEN, or LIST statements. 4) *Patcher* - allows the user to display any sector of a given file or program, and then to update any data within that sector. A second option enables the user to specify the particular sector he wishes to update. 5) *Screen Printer* - permits contents of the screen to be sent to the printer at any time the keyboard is active. The program remains in effect until you press RESET or reboot the system. 6) *Array Editor* - a simple word processor that allows you to create, modify, print and save your own text files. 7) *Calculator II* - a multiplication/division subroutine that handles numeric string data. Written in Assembler code, and using twenty place accuracy, it runs much faster than an equivalent BASIC subroutine. It is also compatible with the addition/subtraction subroutine, the Calculator, included in the first Dakin5 Programming Aids package reviewed in the December 1979 issue of The MICRO Software Catalogue XV.

Copies: **Just released**
Price: **\$49.95**
Includes: **Professionally**

Author: **Dakin5 Corporation**
(developer of The Controller for Apple Computer, Inc.)
Available: **Local Apple Dealers**

Name: **Page Format TTY IN/OUT**
System: **Apple II**
Memory: **300.3FF (256 Bytes)**
Language: **Machine**
Hardware: **Game Conn to TTY**

Description: Program to output to and input from ASK 33 or 35 Teletype. Gives multiple line feeds at end of each page and waits for you to tear off roll paper or insert new sheet for neat listings. Uses game connector.

Copies: **Just released**
Price: **\$2.00**
Includes: **Listing and Instructions**
Author: **Ken Ellis**
Available: **Ken Ellis**
R.D.8 Box 344
York, PA 17403

Name: **HI-RES GRAPHIC CHARTS GENERATOR**
System: **APPLE II, APPLE II PLUS**
Memory: **32K without ROM card, 16k with card**
Language: **APPLESOFT II BASIC**
Hardware: **APPLE II, Disk II (allows optional features)**

Description: This program will allow you to generate HI-RES graphic charts, either through keyboard or text file input (it uses disk). 'Y' axes will be automatically

scaled with values. 'X' axes will be marked for plotting points. Best of all, once graph is automatically created, you can add your own titles, comments, or symbols anywhere on the graph. Both upper and lower case characters are provided. Over 30 special symbols are included. Provisions are also included for multiple graph overlays. Disk II users can automatically have graphs made from existing data already stored.

Copies: **Just releases, 42 copies already sold.**
Price: **\$19.95 + \$1.25 for postage.**
Includes: **Cassette containing program, instructions on unique uses. Please specify when your order, if you have ROM card or not.**

Author: **Les Stubbs**
Available: **Les Stubbs**
23725 Oakheath Pl.
Harbor City, Ca. 90710

Name: **General Ledger Version 2.0**
System: **Apple II**
Memory: **48K**
Language: **Applesoft**
Hardware: **Dual Drives, Any Printer**

Description: General Ledger Version 2.0 — This program is a complete double-entry accounting system. User defined flexibility allowing up to 9 individualized departments in all Financial Reports. 10 levels of subtotals throughout each report gives more detailed Financial Statements. Using 5" drives, storing the entire Chart of Accounts and/or all posting approaches minicomputer times when verifying account numbers or sor-

ting records. High-speed printer routines will process 1,000 postings into 70 accounts in less than 30 minutes. Using 8" drives, high-speed sorting routines requiring no additional disk work space and fast binary searching techniques allow data files to be limited only by your available disk space. Compatible with any printer and printer interface.

Copies: **Version 1.0, 200; Version 2.0, Just released.**
 Price: **\$180.00**
 Author: **David A. McFarling**
 Available: **Small Business Computer Systems 4140 Greenwood Lincoln, NE 68504**

Name: **VDCAB 1.1**
 System: **APPLE II or APPLE II PLUS**
 Language: **Applesoft**
 Memory: **32K**
 Hardware: **APPLE II and Disk II**

Description: A vocabulary builder with over 1200 multiple choice questions allows the user to select either synonyms or antonyms. Intended as study aid for college board type exams (e.g., SAT, ACT, GRE, LSAT, etc.). Editor is included for expanding or modifying data lists. Several test formats with grading are options. Ideal for students with little computer experience.

Price: **\$15.00**
 Includes: **User documentation and diskette**
 Author: **Steven M. Sliwa**
 Available: **Sliwa Enterprises 257 C Clemwood Parkway Hampton, VA 23669**

Name: **SORT**
 System: **PET, APPLE**
 Memory: **32K/16K PET; any Apple**
 Language: **6502 Machine Language**
 Hardware: **16K/32K PET, any APPLE**

Description: SORT is a 6502 machine language intelligent sort for commercial applications. Requires almost no user set-up when default values are used. Sorts integer, string and floating point arrays of more than one dimension with up to 20 sub-sorts-on-match (if needed).

Copies: **Just released**
 Price: **depends on end use**
 Author: **David B. Black**
 Available: **MATRIX SOFTWARE INC. 1041 N. Main St. Ann Arbor, MI 48104**

Name: **Investment Comparison**
 System: **Apple II or Apple II Plus**
 Memory: **32K with RDM Applesoft, 48K with RAM Applesoft**
 Language: **ROM Applesoft. Can be used with RAM Applesoft by re-locating above HGR2 display area or not using graphics display feature. 1024 bytes of Machine Code is loaded before Main program.**
 Hardware: **Cassette tape. Program supports Printer but driver subroutine not included. Apple II.**

Description: We are often faced with decisions such as 'which of two investments is best?' This program provides a means of comparing them by the use of "Cash Discounting." Cash Discounting is a technique that is used to take into consideration the effects of inflation. Often we are faced with a decision of 'buying now' vs waiting a few years or paying cash vs time payments. The effects of inflation are not easy to quantify without some form of computer analysis. For each of two alternatives, entry include:
 1. Inflation Rate for both
 2. Initial Investment \$
 3. Number of years to salvage point and value at that time.
 4. Monthly expenses (or income)
 5. Adjustments on an annual basis for the monthly expenses. This provides a means whereby you can make expenses track at a different rate than inflation. Display is in a form of a 'cash flow' by year, and a graphical presentation is also provided. The graphics have labels.

Copies: **Just released**
 Price: **\$16.95**
 Includes: **Cassette, loading instructions, description, and example.**
 Author: **Neil A. Robin**
 Available: **TECH-DIGIT 21 Canter Lane Sherwood, OR 97140**

Name: **The Life Dynamic Transformation Experience**
 System: **Apple II**
 Memory: **48K**
 Language: **Applesoft and Machine Language**
 Hardware: **Apple II Plus, Disk II**

Description: Unique! This program is designed for all those people who desire to experience self-transformation, life-awareness, making relationships work,

and "getting your act together," but do NOT desire to pay est or Lifespring or any of the other "trips" of the Human Potential Movement, \$300 or so. Includes game playing as a means to a fun way of increasing awareness.

Copies: **Many**
 Price: **\$15.95**
 Includes: **(disk) w/instructions**
 Author: **Avant-Garde Creations**
 Available: **Avant-Garde Creations P.O. Box 30161 Dept. MC Eugene, OR 97403**

Name: **I CHING**
 System: **Apple II or Apple II Plus**
 Memory: **16K**
 Language: **Integer Basic or Applesoft (please specify)**
 Hardware: **Cassette or disk**

Description: Have your own oracle in your home. Consult the I Ching as others have through the ages. Includes a tutorial and a bibliography, as well as an interpretation of the results.

Copies: **Just released**
 Price: **\$9.95 on cassette; \$14.95 on disk**
 Author: **C. Brandon Gresham, Jr.**
 Available: **Ad Hoc Enterprises 23 Van Buren Street Dayton, OH 45402**

Name: **MUSIC**
 System: **Any 6502 based system**
 Memory: **1.5K**
 Language: **Assembly**
 Hardware: **Terminal or TVT and a speaker connected to one output port**

Description: Music is an interactive programming language for the creation of patterns of sound; "music". It is a compositional tool, not merely a music table compiler or piano roll type of program. Music's language structure is similar to "ROBOT" (see MICRO no. 10, page 15). Complex hierarchies of user defined functions - strings of musical events - which can be called like subroutines, allow the user to program highly intricate and surprising compositions.

Copies: **Just released**
 Price: **\$10.00 (KIM-1 Hyper-tape cassette: \$3.00 extra)**
 Includes: **User manual with programming examples and a completely commented source and object code listing**
 Author: **Michael Allen**
 Available: **Michael Allen 6025 Kimbark Chicago, IL 60637**

JOIN RAYGAMCO NOW!

Become a
member of
RAYGAMCO
Computer
Discount Club.



**SAVE
20%
AND MORE!**

BIG SAVINGS ON EVERY ITEM!

By being a RAYGAMCO Member you receive substantial discounts on every item you purchase, including all hardware, software, accessories, even books and paper! You will also receive a monthly newsletter with all the latest available for your particular computer system, and much, much more — exclusive to RAYGAMCO Members only!

All the famous brand names, including:

APPLE	Alpha Micro	Soroc	Lear Siegler
ATARI	Alpha Pro	Hazeltine	Shugart
EXIDY/Sorcerer	Cromemco	Sektor	Texas Instruments
Kim/Commodore	Xerox	PET	

SAVE 20% AND MORE!

Here's how to join.

Fill out the information, and mail. That's all there is to it. Nothing to buy. I want to be a RAYGAMCO Computer Discount Club Member. Please send my RAYGAMCO Membership card to:

Name

Address

City State Zip

Computer (Brand Name)

I would like information on (please specify system, part, accessory, book, program, etc.)

WE HONOR VISA, MASTERCHARGE, BANKAMERICARD.

TOLL FREE, EXCEPT CA

Store Hours: Sat 10-6, Sun 12-4, Tu-Fri 11-8

800-854-6455

RAYGAM, INC.

6791 WESTMINSTER AVENUE WESTMINSTER, CA 92683
TELEX 182274 (714) 891-2587



Ask for Instant Software at a computer store near you.

Alabama

Computerland of Huntsville
3020 University Dr., Huntsville

Arizona

Ham Shack
4506-A N. 16th St., Phoenix
Millete TV & Radio
621 East Broadway, Mesa

California

Byte Shop
8038 Clearmont Mesa Blvd.,
San Diego
Byte Shop at MI View
1415 West El Camino Real, MI View
Capitol Computer Systems
3396 El Camino Ave., Sacramento
Computer Made Easy
819 East Ave. Q-9, Palmdale
Computer World
6791 Westminster Ave., Westminster
Computerland
16720 S. Hawthorne, Lewndale
Coast Electronics
3118 N. Main St., Marina Bay
Computerland
24001 Via Fabricante No 904,
Mission Vieja
Habitronics
1378 S. Baer Ave., San Jose
Hobby World
19511 Buena Vista Cir., Unit 6
Bortherda
Jada Computer Products
4901 W. Rosecrane, Hawthorne
Opamp/Technical Books
1033 N. Sycamore Ave., Los Angeles

Radio Shack Dealer
8250 Mira Mesa Blvd., San Diego
Santa Rosa Computer Center
604 7th St., Santa Rosa
Silver Spur Elect. Comm.
13552 Cantrel Ave., Chino
The Computer Store
820 Broadway, Santa Monica
Colorado
Byte Shop
3464 S. Acacia St., Englewood
Computerland of North Denver
8749 Wadsworth Blvd., Arvada
The Computer Store
2300 Welton St., Denver
Connecticut
Computer Works
1439 Post Rd. E., Liberty Plaza,
Westport
D.C.
The Program Store
4200 Wisconsin Ave. N.W.,
Washington, D.C.

Florida
Adventure International
200 Bald Cypress Ave., Langwood
AMF Electronics
11145 N. 30th St., Tampa
Computerland of Ft. Lauderdale
3963 N. Federal Hwy., Ft. Lauderdale
Computerland of Jacksonville
2777-6 University Blvd. W.,
Jacksonville
Curtis Walters Enterprises
236 Talbot Ave., Melbourne
Ukutan Computer Store
Airport Rd., Destin
Williams Radio & TV Inc.
2062 Liberty St., Jacksonville

Georgia
Computerland of Atlanta
2423 Cobb Parkway, Smyrna
Hawaii
Computerland of Hawaii
567 N. Federal Hwy., Honolulu
Radio Shack Assoc. Store
1712 S. King St., Honolulu
Illinois
Computerland
4507 North Sterling, Peoria
Midwest Micro Computers, Inc.
708 S. Main St., Lombard
Indiana
Computer Center of South Bend
51591 US 31 North, South Bend

Iowa

Memory Bank
4128 Brady St., Davenport

Louisiana

Computer Shoppe Inc.
3225 Dennis Park, Suite 222, Metairie

Maine

Radio Shack
315 Main Mall Rd., Sa. Portland

Massachusetts

ComputerCity
175 Main St., Charlestown
ComputerCity
50 Worcester Rd., Framingham
Computerland at Baelon
214 Watcaet Rd., Wellaalay
Computer Packagaa Unlimited
244 W. Baylata St., Waal Baylata
Lighthouse Computer Software
14 Fall River Ave., Rahobath
The Computer Store
120 Cambridge St., Burlington
Tulsa Radio & Electronics
208 Mystic Ave., Medford
Michigan
Computer Center
28251 Ford Rd., Garden City
Computerland of Grand Rapids
2927 28th St. S.E., Kalamazoo
Computerland at Rochester
301 S. Livernois, Rochester
Computerland at Southfield
29673 Northwestern Hwy., Southfield
Computer Mart
560 W. 14 Mile Rd., Clawean
Hobby House
1035 W. Tarrillan Rd., Battle Creek

Minnesota

Computerland at Hopkins
11319 Hwy F., Hopkins
Minnesota Software Inc.
5422 Fiehar St., White Bear Lake
Mississippi
Dyer's, Inc.
200 E. Main St., West Point
Missouri
Computerland, Inc.
51 Fairleeant Oaks Shopping Center
Florissant
Computerland Software
16501 Greenwald Court, Belton
Montana
Intermountain Computer
529 S. 9th St., Livingston
Personal Computer
121 Red Oak Dr., Carl Junction
The Computer Store
1216 18th St. W. #35, Billings
Nebraska
Computerland at Omaha
11031 Elm St., Omaha
Midwest Computer Co. Inc.
8625 I St., Omaha
Midwest Computer Co. Inc.
4442 S. 84th St., Omaha
Midwest Computer Co. Inc.
4403 S. 87th St., Omaha

New Hampshire

Bitsnbytes Computer Center
568 Pleasant St., Concord
ComputerCity
1525 S. Willow, Manchester
Portsmouth Computer Center
31 Reynes Ave., Portsmouth
New Jersey
Computerland
35 Plaza Rte. #4, W. Paramus
Computer Mart of NJ
501 Rte. 27, Iaelin
Deva's Electronics
Penneville Shopping Ctr., Pennsville
Laahen Electronics Inc.
21 Broadway, Denville
Peraonal Computing Inc.
51 Central Sq., Linwood
The Computer Emporium
Bldg. 103, Avenue of Commerce
2428 Rte. 38, Cherry Hill
New Mexico
Antel Electronics Co.
232 Wisconsin N.E., Albuquerque

New York

Aristo Craft
314 Fifth Ave., NYC
Computer Corner
200 Hamilton Ave., White Plains
Computer Eie Corp.
1570 3rd Ave., New York
Computerland at Naesau
79 Waabury Ave., Carle Place
Computer World
519 Baelan Post Rd., Pari Cheeter
Comtak Electronics, Inc.
2666 Caney Iseland Ave., Brooklyn
Comtek Electronics, Inc.
Stelan Iseland Mall
Stare 226A, Stelan Iseland
Hama Computer Center
671 Manroe Ave., Rochester
Key Electronics
Schenacledy
Mr. Computer
Imp. Plaza, Rte. 9, Wappingers Falls
Soltran Systems
308 Columbia Turnpike, Renaealaar
The Computer Tie Inc.
409 Haaper Rd., Endwall
Upstate Computer Shop
529 Franch Rd., Campus Plaza
New Hartford
Ohio
Altair Buainee Systems, Inc.
5252 North Dixie Dr., Dayton
Aetra Video Electronics
504 E. Main St., Lancaster
Computerland
4579 Graal Northern Blvd.,
N.Olmstead
Computerland
6429 Buach Blvd., Calumbue
Computerland
1288 Sam Rd., Mayfield Haight
Farbes Microsystems Inc.
35 N. Broad, Fairham

Oklahoma

Microcomputer Center
7900 Peragan Rd., Deyton
Micro-Mini Computer World
74 Robinwood, Calumbua
Universal Amateur Radio, Inc.
1280 Aida Dr., Calumbua
Oregon
Computerland at Portland
12020 S.W. Main St., Tigard
Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem
Pennsylvania
Audio Merl
518 Fifth Ave., New Brighton
Arica Elect
302 Wyaming Ave., Kingatan
Arica Elect
Back Mountain Shop. Ctr.
Shavertown
Computer Workshappe
3848 William Penn Hwy. Monroeville
Computerland at Harrisburg
4644 Carlisle Pike, Mechanicsburg
Erie Computer Co.
2127 West 8th St., Erie
Mighty Byte Computer Center
537 Eatan Rd., Heraham
Personal Computer Corp.
24-26 West Lancaster Ave., Peall
Personal Computer Corp.
Frazar Mall, Lancaster Ave., Frazer

Rhode Island

Computer City
165 Angell St., Providence

Tennessee
Computerlab
671 S. Menden Hall Rd., Memphis
H & H Electronics Inc.
509 N. Jackson St., Tullahoma
Texas
Computer Post
926 N. Galling, Arlington
Houston Microcomputer Tech.
5313 Braaanel, Bellaire
K.A. Elect.
9090 Stemmane Frwy., Dallas
Pan American Elect. Inc.
1117 Conway, Mission
Utah
Quality Technology
470 E. 2nd St., Salt Lake City
Virginia
Hame Computer Center
2627 Virginia Beach Blvd
Virginia Beach
Southalder Redia Comm
135 Pickwick Ave., Calaniet Haight
Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Oregon
Computerland at Portland
12020 S.W. Main St., Tigard
Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem
Pennsylvania
Audio Merl
518 Fifth Ave., New Brighton
Arica Elect
302 Wyaming Ave., Kingatan
Arica Elect
Back Mountain Shop. Ctr.
Shavertown
Computer Workshappe
3848 William Penn Hwy. Monroeville
Computerland at Harrisburg
4644 Carlisle Pike, Mechanicsburg
Erie Computer Co.
2127 West 8th St., Erie
Mighty Byte Computer Center
537 Eatan Rd., Heraham
Personal Computer Corp.
24-26 West Lancaster Ave., Peall
Personal Computer Corp.
Frazar Mall, Lancaster Ave., Frazer

Rhode Island
Computer City
165 Angell St., Providence

Tennessee
Computerlab
671 S. Menden Hall Rd., Memphis
H & H Electronics Inc.
509 N. Jackson St., Tullahoma
Texas
Computer Post
926 N. Galling, Arlington
Houston Microcomputer Tech.
5313 Braaanel, Bellaire
K.A. Elect.
9090 Stemmane Frwy., Dallas
Pan American Elect. Inc.
1117 Conway, Mission
Utah
Quality Technology
470 E. 2nd St., Salt Lake City
Virginia
Hame Computer Center
2627 Virginia Beach Blvd
Virginia Beach
Southalder Redia Comm
135 Pickwick Ave., Calaniet Haight
Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Oregon
Computerland at Portland
12020 S.W. Main St., Tigard
Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem
Pennsylvania
Audio Merl
518 Fifth Ave., New Brighton
Arica Elect
302 Wyaming Ave., Kingatan
Arica Elect
Back Mountain Shop. Ctr.
Shavertown
Computer Workshappe
3848 William Penn Hwy. Monroeville
Computerland at Harrisburg
4644 Carlisle Pike, Mechanicsburg
Erie Computer Co.
2127 West 8th St., Erie
Mighty Byte Computer Center
537 Eatan Rd., Heraham
Personal Computer Corp.
24-26 West Lancaster Ave., Peall
Personal Computer Corp.
Frazar Mall, Lancaster Ave., Frazer

Rhode Island
Computer City
165 Angell St., Providence

Tennessee
Computerlab
671 S. Menden Hall Rd., Memphis
H & H Electronics Inc.
509 N. Jackson St., Tullahoma
Texas
Computer Post
926 N. Galling, Arlington
Houston Microcomputer Tech.
5313 Braaanel, Bellaire
K.A. Elect.
9090 Stemmane Frwy., Dallas
Pan American Elect. Inc.
1117 Conway, Mission
Utah
Quality Technology
470 E. 2nd St., Salt Lake City
Virginia
Hame Computer Center
2627 Virginia Beach Blvd
Virginia Beach
Southalder Redia Comm
135 Pickwick Ave., Calaniet Haight
Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Oregon
Computerland at Portland
12020 S.W. Main St., Tigard
Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem
Pennsylvania
Audio Merl
518 Fifth Ave., New Brighton
Arica Elect
302 Wyaming Ave., Kingatan
Arica Elect
Back Mountain Shop. Ctr.
Shavertown
Computer Workshappe
3848 William Penn Hwy. Monroeville
Computerland at Harrisburg
4644 Carlisle Pike, Mechanicsburg
Erie Computer Co.
2127 West 8th St., Erie
Mighty Byte Computer Center
537 Eatan Rd., Heraham
Personal Computer Corp.
24-26 West Lancaster Ave., Peall
Personal Computer Corp.
Frazar Mall, Lancaster Ave., Frazer

Rhode Island
Computer City
165 Angell St., Providence

Tennessee
Computerlab
671 S. Menden Hall Rd., Memphis
H & H Electronics Inc.
509 N. Jackson St., Tullahoma
Texas
Computer Post
926 N. Galling, Arlington
Houston Microcomputer Tech.
5313 Braaanel, Bellaire
K.A. Elect.
9090 Stemmane Frwy., Dallas
Pan American Elect. Inc.
1117 Conway, Mission
Utah
Quality Technology
470 E. 2nd St., Salt Lake City
Virginia
Hame Computer Center
2627 Virginia Beach Blvd
Virginia Beach
Southalder Redia Comm
135 Pickwick Ave., Calaniet Haight
Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Oregon
Computerland at Portland
12020 S.W. Main St., Tigard
Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem
Pennsylvania
Audio Merl
518 Fifth Ave., New Brighton
Arica Elect
302 Wyaming Ave., Kingatan
Arica Elect
Back Mountain Shop. Ctr.
Shavertown
Computer Workshappe
3848 William Penn Hwy. Monroeville
Computerland at Harrisburg
4644 Carlisle Pike, Mechanicsburg
Erie Computer Co.
2127 West 8th St., Erie
Mighty Byte Computer Center
537 Eatan Rd., Heraham
Personal Computer Corp.
24-26 West Lancaster Ave., Peall
Personal Computer Corp.
Frazar Mall, Lancaster Ave., Frazer

Rhode Island
Computer City
165 Angell St., Providence

Tennessee
Computerlab
671 S. Menden Hall Rd., Memphis
H & H Electronics Inc.
509 N. Jackson St., Tullahoma
Texas
Computer Post
926 N. Galling, Arlington
Houston Microcomputer Tech.
5313 Braaanel, Bellaire
K.A. Elect.
9090 Stemmane Frwy., Dallas
Pan American Elect. Inc.
1117 Conway, Mission
Utah
Quality Technology
470 E. 2nd St., Salt Lake City
Virginia
Hame Computer Center
2627 Virginia Beach Blvd
Virginia Beach
Southalder Redia Comm
135 Pickwick Ave., Calaniet Haight
Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Oregon
Computerland at Portland
12020 S.W. Main St., Tigard
Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem
Pennsylvania
Audio Merl
518 Fifth Ave., New Brighton
Arica Elect
302 Wyaming Ave., Kingatan
Arica Elect
Back Mountain Shop. Ctr.
Shavertown
Computer Workshappe
3848 William Penn Hwy. Monroeville
Computerland at Harrisburg
4644 Carlisle Pike, Mechanicsburg
Erie Computer Co.
2127 West 8th St., Erie
Mighty Byte Computer Center
537 Eatan Rd., Heraham
Personal Computer Corp.
24-26 West Lancaster Ave., Peall
Personal Computer Corp.
Frazar Mall, Lancaster Ave., Frazer

Rhode Island
Computer City
165 Angell St., Providence

Tennessee

Computerlab
671 S. Menden Hall Rd., Memphis
H & H Electronics Inc.
509 N. Jackson St., Tullahoma

Texas

Computer Post
926 N. Galling, Arlington
Houston Microcomputer Tech.
5313 Braaanel, Bellaire
K.A. Elect.
9090 Stemmane Frwy., Dallas
Pan American Elect. Inc.
1117 Conway, Mission
Utah
Quality Technology
470 E. 2nd St., Salt Lake City
Virginia
Hame Computer Center
2627 Virginia Beach Blvd
Virginia Beach
Southalder Redia Comm
135 Pickwick Ave., Calaniet Haight
Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Utah

Quality Technology
470 E. 2nd St., Salt Lake City

Virginia

Hame Computer Center
2627 Virginia Beach Blvd
Virginia Beach
Southalder Redia Comm
135 Pickwick Ave., Calaniet Haight
Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Washington

American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Magnolia Micro Systems
2912 Thornadya Ave., Seattle
Personal Computer
5104 Freva, Spokane
Ya Old Computer Shop
1301 G. Washington, Richland
Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

Wisconsin

Byte Shop Of Milwaukee
6019 West Layton Ave., Grantfield
Pellet Microsystems
4265 W. Laornie Rd., Milwaukee

If the store nearest you does not stock Instant Software, use this order blank to purchase your software directly, or call Toll-Free 1-800-258-5473.

Name _____				
Address _____				
City _____		State _____		Zip _____
<input type="checkbox"/> Check	<input type="checkbox"/> Money Order	<input type="checkbox"/> VISA	<input type="checkbox"/> AMEX	<input type="checkbox"/> Master Charge
Card No. _____		Expiration Date _____		
Signed _____		Date _____		
Order your Instant Software today!				
Quantity	Order No.	Program name	Unit cost	Total cost
Shipping and handling				\$1.00
Total order				

Instant Software™ Inc.

Peterborough, N.H. 03458

Instant Software™ Inc.

Peterborough, New Hampshire 03458
603-924-7296

Get a load of this . . . Instant Software™

Ask for Instant Software at a computer store near you.

PET UTILITY I You're working under a serious handicap if you can't write programs in machine language. The PET Utility I package gives you the tools you need:

- **Monitr**—The Monitr program lets you write, edit, save, and verify any machine language and/or BASIC program. Just load and run the Monitr program and then load the program you want to edit.

- **Programmer's Calculator**—This program will convert numbers into the binary, octal, decimal, and hexadecimal systems and function as a floating point calculator. It will also display all four numbering systems simultaneously and allow you to handle large numbers limited only by the size of your screen. For the 8K PET. Order No. 0105P. \$7.95.



ACCOUNTING ASSISTANT This package will help any businessman solve many of those day-to-day financial problems. Included are:

- **Loan Amortization Schedule**—This program will give you a complete breakdown of any loan or investment.

- **Depreciation Schedule**—You can get a depreciation schedule using any one of the following methods: straight line, sum of years-digits, declining balance, units of production, or machine hours.

This package is available for both the PET and Apple. It requires the Apple 16K and Apple-Soft II BASIC. Order No. 0088A \$7.95 or the PET 8K. Order no. 0048P \$7.95.

MIMIC Test your memory and reflexes with the live different versions of this game. You must match the sequence and location of signals displayed by your Apple. You'll need an Apple with 24K and Integer BASIC. Order No. 0025A \$7.95.

HAM PACKAGE I This versatile package lets you solve many of the problems commonly encountered in electronics design. With your 8K PET, you have a choice of:

- **Basic Electronics with Voltage Divider**—Solve problems involving Ohm's Law, voltage dividers, and RC time constants.

- **Dipole and Yagi Antennas**—Design antennas easily, without tedious calculations. This is the perfect package for any ham or technician. Order No. 0054P \$7.95.

SAHARA WARRIORS Enjoy all the gritty realism of desert warfare:

- **Commando**—You must send your commandoes to trap a German general and cut him off from his troops.

- **French Foreign Legion**—The battalions of the French Foreign Legion are in a race with the Arabs. Which side will get its battalions into the oasis in the shortest time? You and a friend can find out.

Both programs in this package require an Apple 8K and Integer BASIC. Order No. 0080A \$7.95.

MORTGAGE WITH PREPAYMENT OPTION/ FINANCIER These two programs will more than pay for themselves if you mortgage a home or make investments:

- **Mortgage with Prepayment Option**—Calculate mortgage payment schedules and save money with prepayments.

- **Financier**—Calculate which investment will pay you the most, figure annual depreciation, and compute the cost of borrowing, easily and quickly.

All you need to become a financial wizard with a 16K Apple and Integer BASIC. Order No. 0094A \$7.95.

CHIMERA If you think the legendary Chimera was hard to handle, wait until you try this package. Included are:

- **Dropotts**—You must make your opponent's men "drop off" the board by moving and firing your own men. For one or two players.

- **Dots**—Place your lines carefully as you try to build and capture the squares. For one player.

- **Battar-up**—You and another player take turns at bat as your PET becomes both the pitcher and the umpire. For two players.

- **Rettex**—Round and round the little white ball rolls. Only fast reflexes can guide it into the center of the maze.

You'll almost be able to feel the Chimera's fiery breath as you play the games on your 8K PET. Order No. 0110P. \$7.95.



CODE NAME: CIPHER Enjoy that same feeling of intrigue and discovery with the Code Name: Cipher package. Included are:

- **Memory Game**—Would you like to match your memory against the computer's? You can with the Memory Game.

- **Codemaster**—One player types in a word, phrase, or sentence, and the PET translates that message into a cryptogram. The other player must break the code and solve the cryptogram in the shortest time possible.

- **Deceitful Mindmaster**—This isn't your ordinary Mastermind-type game. You must guess the live letters in the hidden code word.

- **Code Breaker**—Cracking this code won't be as easy as cracking walnuts. You'll need to flex your mental muscles to win this game.

If you want a mental challenge, then Code Name: Cipher is for you. For the 8K PET, Order No. 0112P. \$7.95.

Ask your local dealer for the latest Instant Software catalog or write: Instant Software Catalog Dept., Peterborough, N.H. 03458

Copyright 1980 by Instant Software Inc.
Peterborough, New Hampshire 03458

All Rights Reserved

Instant Software Inc.

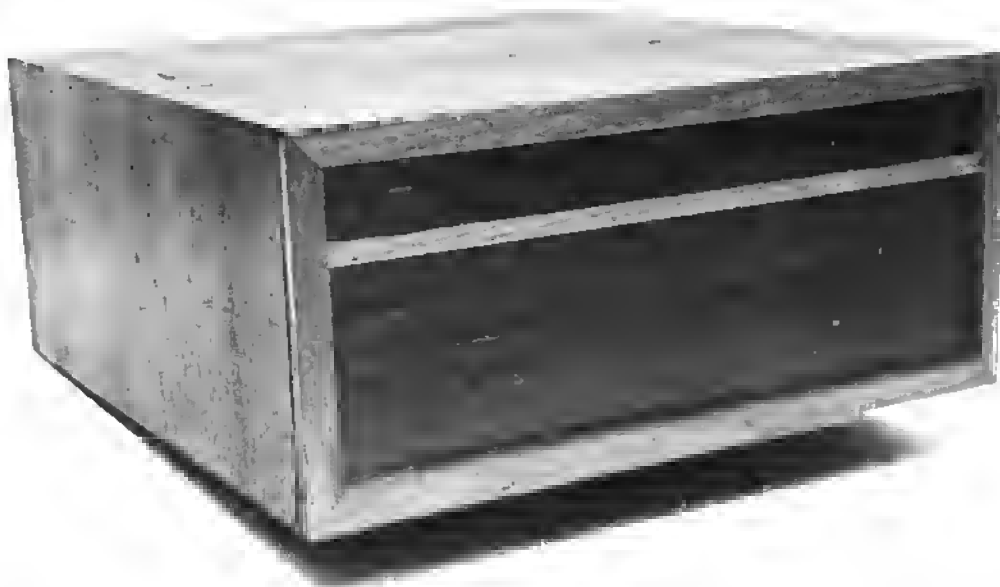
Peterborough, New Hampshire 03458 603-924-7296



BOX 120
ALLAMUCHY, N.J. 07820
201-362-6574

HUDSON DIGITAL ELECTRONICS INC.

ANNOUNCING THE HDE OMNIDISK 65/8



Now, you can "plug in" the latest in a successful series of flexible disk systems developed by HDE for the KIM, SYM and AIM microcomputers. The OMNIDISK 65/8 is a complete system, using 8 inch soft sectored diskettes with a formatted (IBM Standard) capacity of 256K. Of course, a disk formatting function is included as are system supporting utilities for file renaming, disk packing, copy (dual systems) and others.

TED, a full featured, line oriented editor is standard in KIM and SYM based versions to get you up and running on your project in a hurry. The AIM version uses the on-board editor. With the OMNIDISK 65/8 you can con-

centrate on your problem, the disk supports you all the way.

OMNIDISK 65/8 is available in an attractive walnut wood cabinet, or unpackaged for OEM applications in dual and single drive configurations. The HDE disk controller is a state-of-the-art 4½" by 6½" card electronically compatible with the 44-pin KIM-4 bus structure. The controller and disk-driver are designed to operate with the popular Shugart 801-R and compatible devices.

The OEM single drive is \$1195, the dual, \$1895 and the dual in the walnut cabinet, \$2200. Price is another reason to step up to the proven quality of an HDE system.

**HDE PRODUCTS - BUILT TO BE USED WITH CONFIDENCE
AVAILABLE DIRECT OR FROM THESE FINE DEALERS:**

JOHNSON COMPUTER
Box 523
Medina, Ohio 44256
216-725-4560

ARESCO
P.O. Box 43
Audubon, Pa. 19407
215-631-9052

PLAINSMAN MICROSYSTEMS
Box 1712
Auburn, Ala. 36830
800-633-8724

LONE STAR ELECTRONICS
Box 488
Manchaca, Texas 78652
612-282-3570

PERRY PERIPHERALS
P.O. Box 924
Miller Place, N.Y. 11764
516-744-6462

6502 Bibliography: Part XIX

Dr. William R. Dial
438 Roslyn Avenue
Akron, OH 44320

581. Compute, Iss. 1 (Fall 1979)

- Moser, Carl W, "Universal 6502 Memory Test," pgs. 32-33.
A memory test program with ways to adapt it to PET, APPLE II, SYM, KIM, TIM, OSI 65D, Western Data Systems, ATARI, AIM, Super Kim, etc...
- Thornburg, Katie A. and David D., "Elying with PET PILOT," pgs. 40-45.
Kids and microcomputers at Peninsula School.
- Tulloch, Michael, "CORVUS 11A Disc Drive," pg. 61.
How about a 9.6 megabyte memory on line? Compatible with the Apple DOS. Plugs Into one slot of the Apple II.
- Victor, John, "Atari Computers: The Ultimate Teaching Machines?," pgs. 62-64.
Discussion of the advantages of the Atari in educational use.
- Lindsay, Len, "The Evolution of a Magazine," pgs. 65-66.
Discusses the history of the PET Gazette and its successor.
- Butterfield, Jim, "PET In Transition," pgs. 68-70.
Discusses new modifications of the PET and how programs must be modified to accomodate the new systems.
- Isaacs, Larry, "Retrotitting ROMs," pgs. 76-77.
How to replace the old PET Roms with the new units.
- Malmberg, David, "Screen Print Routine," pgs. 78-79.
General utility to print the screen using the new PET printers.
- Anon., "Cassette Format Revisited," pgs. 80-81.
Discussion of the PET Cassette format.
- Butler, Brett, "Trace for the PET," pgs. 84-85.
TRACE allows you to see Basic executing, displaying each actual line as it is executed.
- Hunkins, Arthur, "8-Bit Digital to Analog Converter," pgs. 90-91.
A review of the MicroTechnology Unlimited DAC board for Hal Chamberlin's 4-part music program.
- Stuart, Chuck, "Using Direct Access Files with the Dual Drive Disk," pgs. 93-96.
A tutorial on the Commodore 0240 Dual Drive Disk.

582. Creative Computing 5, No 10 (October 1979)

- North, Steve, "Mountain Hardware SUPERTALKER," pgs. 42-44.
A review of this new accessory for the Apple II.

- Yob, Gregory, "Personal Electronic Transactions," pgs. 180-183.
Changes in the New PET; more on the structure of Basic variables; Simple variables, PET Basic Pointers, Discussion of cryptograms.

583. The Target (Sept/Oct 1979)

- Roland, Don, "AIM 65 MOVIT," pgs. 2-3.
A move program for the AIM 65.
- Butterfield, Jim, "Mortgage," pg. 4.
Mortgage program for AIM Basic.
- Clem, Don, "Memory Display," pg. 5.
A program to show what the 6502 is seeing at the output port of the AIM micro.
- Riley, Ron, "AIM BASIC," pgs. 6-8.
All about AIM Basic.
- Anon., "Statistical Analysis," pg. 10.
Regression programs for the AIM.

584. The Paper 2, Iss. 7 (September 1979)

- Busdiecker, Roy, "Think Negative!," pgs. 3-7.
Negative numbers and subtraction are covered in this continuing tutorial on binary numbers, with examples for the PET and 6502.
- Busdiecker, Roy, "Decimal to Binary," pg. 12.
PET Basic program for converting Decimal to Binary.
- Busdiecker, Roy, "Warning: Prevent Tape Decay," pgs. 21-22.
Use your PET tapes or at least rewind them occasionally to avoid print through on long undisturbed storage.
- Oakes, Peter L. A., "PLOT 2-MP," pgs. 23-25.
A plotting program for the PET.

585. Call Apple 2, No. 7 (September 1979)

- Wagner, Roger, "Speeding in Applesoft," pg. 2.
How to speed up Applesoft Subroutines.
- Winston, Alan B., "The Multi-Lingual Apple," pgs. 4-5.
All about Pascal for the Apple.
- Hertzfeld, Andy, "Assembler Mini Reviews," pgs. 7-10.
A Consumer's guide to Apple II Assemblers: Apple Mini-Assembler, TED/ASM, Micro Products (Moser/Bishop) Assembler, Randy's Weekend Assembler, Aresco

Assembler, LISA, Microproducts 6-Character Label Assembler, E.A.T. (Apple II Text Processing System), 6502 Macro Assembler (Moser), UCSD Pascal Assembler, S-C Assembler, etc. etc....

Golding, Val J., "Peeking at Numbers," pg. 10.

Demonstration of a couple of ways to print numbers in Integer Basic larger than 32767 on the Apple.

Verlaque, Richard, "Decimal Division/Integer Basic," pg. 16.
Arithmetic program for Integer Basic on the Apple.

Thing, Mike, "Applemash," pgs. 18-19.

All about Modem Operation and the newly established "Apple-Crate," the Call-Apple message system ABBS.

Hoyte, Jim, "Subroutine to Allow Prohibited Character in String Inputs," pg. 24.

Routine lets you use commas, etc. in string inputs on the Apple.

Williams, Rick, "Flash Cards," pg. 28.

A Tutorial program for the Apple.

Capella, Mark, "Hide," pg. 29.

A program to hide program names from appearing in the catalog, but still permitting them to be recovered.

Golding, Val J., "Poke Chr\$ to screen," pg. 29.

Routine to put characters on screen at specified locations.

Dunmire, Darrell and Golding, Val J., "So Who Needs Applesoft?" pg. 31.

How to use string functions such as VAL, STR\$, and CHR\$ in Integer Basic on the Apple.

Neulen, Bob and Golding, Val J., "Change Catalog to CC," pg. 33.

A routine to change catalog to CC for the 3.2 DOS.

Corsetti, Vincent S., "File Restore," pg. 35.

How to restore a program on a disk after you realize you just wiped out something you didn't mean to, on the Apple.

586. Personal Computing 3, No. 11 (November, 1979)

Anon., "Jim and Kay Weir Re-invent the Wheel," pg. 14.

Discusses an application of an Apple II in the tire retreading business.

587. Computer Cassettes Review, No 7 (Fall 1979)

Purser, Robert Elliott, "Software for Apple, PET and TRS-80." Software Directory, cassette reviews, game reviews, and a list of "The Classics—the best of the past."

588. Rainbow 1, No. 8 (September 1979)

Ladbroke, Brian D., "Running PIMS on the APPLE II," pg. 1.
How to adapt this TRS-80 or PET program on Personal Information Management System to the Apple II.

Rost, Randi J., "HiRes Color," pgs. 4-7.

A tutorial.

Heller, H. Lewis and Gothrie, Nelson, "Dollar Conversion," pgs. 7-11.

Three programs for the Apple to convert Applesoft outputs to rounded off dollar and cents values.

Buchen, Steven E., "Slow Down," pgs. 16-21.

Two programs to slow down the list function of the Apple MacDougall, John, "Parallel Interface for the Apple," pgs. 22-24.

Hardware article on a parallel interface providing 2 output ports and 2 input ports.

589. Interface Age 4, No. 12 (December 1979)

Gilbert, Betsy, "The Computerized Artist," pgs 72-73.

A description and evaluation of the Apple Graphics Tablet.

590. Kilobaud Microcomputing No. 35 (November 1979)

Lancaster, Don, "Lower case for your Apple II, Part 1," pgs. 30-36.

Expand the usefulness of your Apple with this inexpensive addition.

Schmeltz, Leslie R., "The Apple Goes to Market," pgs. 70-76.
Gather and analyze data from the stock market.

Derfler, Frank J., "Boy, Did I Make a Killing!" pgs. 112-114.
Real Estate profit guide program for the PET.

Bajcz, Bill, "A 'Pentronics' System," pgs. 158-160.
How to interface a PET to a Centronics 101 Printer.

591. Southeastern Software Newsletter No. 13 (Oct. 1979)

McClelland, George, "Pascal," pgs. 1-3.

Discussion of Pascal for the Apple.

Carpenter, Chuck, "Indexing Fundamentals, Part II," pgs. 4-6.
A tutorial on indirect addressing including a routine to read and print a memory range.

Anon., "Four HI-RES Programs," pgs 7-8.
Several hires routines to frame pages of a program.

592. Ohio Scientific's Small Systems Journal (1979)

Anon., "OSI Programs."

A collection of about 60 programs for OSI computers.

593. The Cider Press 2, No. 5 (September 1979)

Anon., "Disk of the Month," pg. 3.

Seventeen more programs.

Anon., "Serial Handshake Modification," pg. 7.

The high speed serial interface card of Apple Computer Company can be made to run faster than 300 baud by simple modification.

Hockenfull, James L., "Betler Sounding Apples," pg. 8.
How to improve the sound of the apple by toggling the cassette output jack.

Nareff, Max J., "Crossfooting," pg. 8.
How to use crossfooting in tabulation of numbers.

594. MICRO 17 (October 1979)

Connolly, Rick, "Nicer Writer," pgs. 5-6.

Eliminate wraparound in your Apple.

Reynolds, William, "Disassembling the DOS 3.2," pgs. 7-10.
Use the Apple DOS 3.2 more effectively with this information on its organization.

Scanlon, C. H., "Hooking PET to Ma Bell," pgs. 11-13.
Use your system as a terminal with an inexpensive modem.

Mimlitch, Thomas R., "Spelunker," pgs. 15-24.
An adventure type game for the Apple.

DeJong, Marvin L., "6522 Timing and Counting Techniques," pgs. 27-39.

Application of the 6522 versatile interface adapter.

Chan Hark, "Card Shuffling Program for KIM-1," pgs. 41-42.
Teach the Kim to shuffle the cards.

Hoyt, Bruce, "How Do You Connect Peripherals to Your Superboard II?" pgs. 43-46.
Some concise information on the configuration and use of the I/O Ports of the OSI Superboard II.

Rowe, Mike (staff), "The MICRO Software Catalog: XIII," pgs. 49-51.

Nine programs are reviewed, mostly for Apples.

Morris, E. D., "Hypocycloids," pgs. 52-53.
A fast graphics program for the OSI.

MUSICAL COMPUTER I AND II

Learn How to Read Music!

Music lessons taught you in your home or at a studio cost from \$7 each half-hour and up! Now, available to you Apple II owners, is an opportunity for you to have your private music teacher in the comfort of your own home.

Written by a M.A. educator with over 20 years of music experience, this two-program cassette provides an *alternative* to music education!

These programs, utilizing high resolution graphics and a 32K Apple II, will make music learning fun and enjoyable for the entire family!

Apple II is a TM of Apple Computers, Inc.

COMPUTER
APPLICATIONS
TOMORROW

P.O. Box 7000-363
Palos Verdes, CA 90274

Name:
Address:
City, State, zip:

Send Check or Money Order for \$34.95 plus \$1.00
postage and handling.
(CA residents add 6% sales tax)

OSI

OSI

SOFTWARE FOR OHIO SCIENTIFIC

Over 50 programs for C1, C2, C4 & Superboard, on tape and disk. All come with listings and complete documentation.

GAMES - 4K - Tape

CHESS FOR OSI - \$19.95
specify system
STARFIGHTER 5.95
Real time space war.
SEAWOLFE 5.95
Floating mines, three
target ships, etc.
LUNAR LANDER 5.95
With full graphics
TEN TANK BLITZ 9.95
A sophisticated real time
tank game.
8K GAMES
BACKGAMMON 9.95
BLACKJACK 6.95
Plays all Vegas rules
Add \$1.00 each for Color/Sound

UTILITIES

C1P CURSOR CONTROL \$9.95
gives reel backspace, one key
screen clear, and **midline editing**
RENUMBERER 5.95
SUPERUTILITY 12.95
Has Renumberer, Variable table
maker and Search
BUSINESS
SMALL BUSINESS ANALYSIS 15.95
Does profit and loss, quick ratio,
break-even analysis and more. 13
pages of documentation.
STOCK PORTFOLIO 6.95
Keeps track of your investments

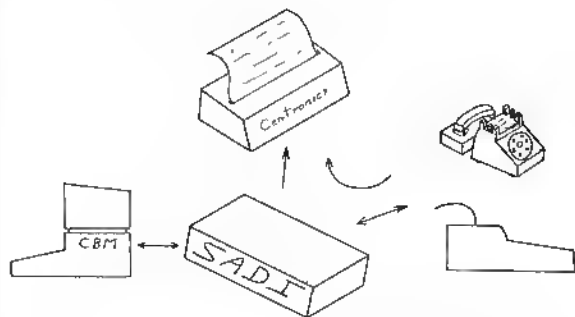
Our \$1.00 catalog has free game
and utility listings, programming
hints and a lot of PEEKs and
POKEs and other stuff that OSI
forgot to mention - and a lot more
programs for sale.

DISKS 5" COLOR/SOUND \$29.95
DISK 1. STARFIGHTER, ROBO-
TANK, SEA WOLFE, BOMBER,
TEN TANK BLITZ
DISK 2 BREAKTHROUGH, LUNAR
LANDER, ALIEN INVADER, KILL-
ERBOTS, SLASHBALL

AARDVARK

1690 Bolton, Walled Lake, Michigan 48088 • (313) 624-6316

PET TWO-WAY RS-232 and PARALLEL OUTPUT INTERFACE



SADI - The microprocessor based serial and parallel interface for the Commodore PET. SADI allows you to connect your PET to parallel and serial printers, CRT's, modems, acoustic couplers, hard copy terminals and other computers. The serial and parallel ports are independent allowing the PET to communicate with both peripheral devices simultaneously or one at a time. In addition, the RS-232 device can communicate with the parallel device.

Special Features for the PET interface include:

- Conversion to true ASCII both in and out
- Cursor controls and function characters specially printed
- Selectable reversal of upper and lower case
- PET IEEE connector for daisy chaining
- Addressable - works with other devices

Special Features for the serial interface include:

- Baud rate selectable from 75 to 9600
- Half or full duplex
- 32 character buffer
- X-ON, X-OFF automatically sent
- Selectable carriage return delay

Special Features for the parallel interface include:

- Data strobe - either polarity
- Device ready - either polarity
- Centronics compatible

Complete with power supply, PET IEEE cable, RS-232 connector, parallel port connector and case. Assembled and tested.

SAD1a (110VAC) \$295

SAD1e (230VAC) \$325

Order direct or contact your local computer store

CONNECTICUT microCOMPUTER, Inc.

150 POCONO ROAD

BROOKFIELD, CONNECTICUT 06804

TEL: (203) 775-9659 TWX: 710-456-0052

VISA AND M/C ACCEPTED - SEND ACCOUNT NUMBER, EXPIRATION DATE AND SIGN ORDER.
ADD \$3 PER ORDER FOR SHIPPING & HANDLING - FOREIGN ORDERS ADD 10% FOR AIR POSTAGE.

STOCK MARKET ANALYSIS PROGRAM DJI WEEKLY AVERAGE 1897-DATE

ANAI (ANALYSIS 1) is a set of BASIC Programs which enables the user to perform analyses on the Dow Jones Industrial weekly average data. From 6 months to 5 years of user selected DJI data can be plotted on the entire screen in one of 5 colors using Apple's High Resolution capabilities. The DJI data can be transformed into different colored graphic representations called transforms. They are: user specified moving averages; a least squares linear fit (best straight line); filters for time, magnitude, or percentage changes; and user created relationships between the DJI data, a transform, or a constant using +, -, x, / operators. Colored lines can be drawn between graphic points. Graphic data values or their dates of occurrence can be displayed in text on the screen. Any graph or text can be outputted to a user's printer. The Grid Scale is automatically set to the range of the graphs or can be user changed. As many colored graphs as wanted can be plotted on the screen and cleared at any time. The user can code routines to operate on the DJI/transform data or create his own disk file data base. ANAI commands can be used with his routines or data base. An Update program allows the user to easily update the DJI file with current DJI weekly data.

The ANAI two letter user commands are: CA = Calculate, no graph. CG = Clear Graphs, leave Grids. CK = Checking out program, known data. CO = Color of next graph (red, green, violet, white, blue). CS = Clear Screen. DL = Draw Line between points. FI = Filter data for time, magnitude, or percent change. FU = Data, transform, or constant function with +, -, x, / operator. GD = Graphic mode, display all Graph Data on screen. GR = Graph data to screen. GS = Set Grid Scale. HE = Help, summary of any commands usage. LD = Load Data from disk file from inputted date to memory. LG = Leave Graphs, automatic Grid rescaling. LO = Look, select a range of the LD data and GR. All commands can now be used on this range. LS = Least squares linear fit of the data. MA = Moving Average of the data. NS = No Scale, next graph on screen does not use Grid Scale. NT = No Trace. PR = User implemented Printer routine. TD = Text mode, display Text Data on screen. TI = Time number to date or vice versa. TR = Trace. TS = Text Stop for number of lines outputted to screen when in TD. U1/U2 = User 1/2 implemented routines. VD = Values of Data outputted in text. VG = Values of Grid: low/high/delta. VT = Values of Transform outputted in text.

APPLE II, 48 K, APPLESOFT
ROM CARD, DISK II DOS 3.2
ANAI DISK & MANUAL ... \$49.95
(CA residents add 6% sales tax)

GALAXY
DEPT. 400
P.O. BOX 22072
SAN DIEGO, CA 92122

* Software Review in Call-A.P.P.L.E. (2/80): "An example of an excellent piece of software exploiting most of Apple II's major features." Overall Rating = 92.1

* Software Review in Apple Orchard (3/80): "A remarkably flexible approach to the analysis and plotting of any time series data." Overall Rating = 85.7

DISCOUNT DATA PRODUCTS

**BASF 5 1/4" DISKETTES:
\$34.50 PER BOX OF 10**

HIGHEST QUALITY DISKETTES AT A
BARGAIN PRICE! LABELS AND WRITE-
PROTECT TABS INCLUDED.

VINYL DISKETTE HOLDERS FOR NOTEBOOKS

THE IDEAL WAY TO STORE DISKETTES. EACH
VINYL PAGE HOLDS TWO DISKETTES AND IN-
CLUDES A POCKET FOR EACH DISKETTE'S LABEL.
SAFELY KEEP UP TO 40 DISKETTES IN A SINGLE
1" 3-RING NOTEBOOK!

\$4.95/SET OF 10

MARKETING YOUR OWN SOFTWARE?
DDP OFFERS DEALER & SOFTWARE HOUSE
DISCOUNTS ON NOT ONLY THE ABOVE
ITEMS, BUT ALSO THE FOLLOWING
PRODUCTS:

9" x 12" ZIP-LOCK BAGS FOR PACKAGING &
DISPLAY OF SOFTWARE.

CORRUGATED MAILERS TO SHIP TO USERS
OR DEALERS!

SEND FOR FREE INFORMATION AT:

DISCOUNT DATA PRODUCTS

P.O. BOX 19674-M

SAN DIEGO, 92119

(ADD \$1.00 SHIPPING/HANDLING CHARGE TO ALL ORDERS.)

KIMEX-1 HERE'S A NEAT COMBINATION

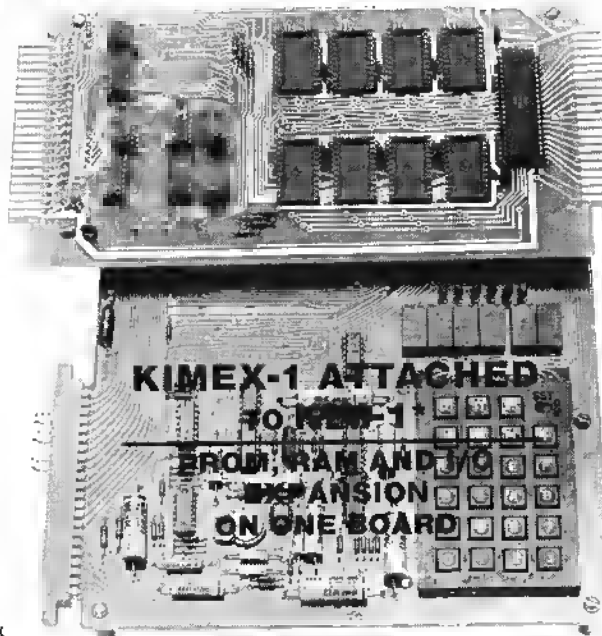
IDEAL FOR DEDICATED INDUSTRIAL OR PERSONAL APPLICATION

FEATURES

- PLUGS DIRECTLY INTO AND COVERS UPPER HALF OF KIM-1. EXPANSION FINGERS CARRIED THROUGH FOR FURTHER EXPANSION.
- I/O-POWERFUL 6522 VIA PROVIDED. (VERSATILE INTERFACE ADAPTER)
- 16 BI-DIRECTIONAL I/O LINES
- 4 INTERRUPT/HANDSHAKE LINES
- 2 INTERVAL TIMERS
- SHIFT REGISTER FOR SERIAL-PARALLEL/PARALLEL-SERIAL OPERATIONS.
- RAM-SOCKETS PROVIDED FOR 4K RAM CONTIGUOUS WITH KIM RAM. (LOW POWER MOSTEK 4118 1KX8's)
- COMPLETE DOCUMENTATION
- EPROM SOCKETS PROVIDED FOR 8K EPROM. (INTEL 2716 2KX8's)
- BLOCK SELECT SWITCHES FOR EPROM. EPROM USABLE IN ANY ONE OF FOUR 8K BLOCKS FROM 8000H.
- AUTOMATIC RESET ON POWER-UP AND SWITCH SELECTABLE INTERRUPT VECTORS.
- PERMITS UNATTENDED OPERATION.
- LOW POWER CONSUMPTION-5V AT 300 Ma. FULLY LOADED
- BUFFERED ADDRESS LINES
- HIGH QUALITY PC BOARD, SOLDER MASK
- ASSEMBLED AND TESTED

APPLICATIONS

PROM, RAM AND I/O EXPANSION ON ONE BOARD HAVING MANY INDUSTRIAL/HOME APPLICATIONS FOR DATA ACQUISITION, PROCESS CONTROL, AUTOMATIC CONTROL OF FURNACE, SOLAR HEAT, LIGHTING, APPLICATIONS, ETC.



THIS IS THE COMPLETE KIMEX-1



PA RESIDENTS INCLUDE 6% STATE SALES TAX

DIGITAL ENGINEERING ASSOCIATES

P.O. BOX 207 • BETHLEHEM, PA 18016

\$139.95

LIMITED TIME 1K RAM **FREE!!!**

* KIM IS A REGISTERED TRADEMARK OF MOS TECHNOLOGY, INC.

NOW PRESENTING...

Software for Apple® II

for your Entertainment · Business · Education

Star Attractions:

+16K Word Perfect Professional Word Processing lets you edit, move, delete, find, change and repeat any body of text, merge and save on disk. Does right-justified margins, centering, page numbering. You can enter name & address onto form letters when printing. Edit and merge any text disk file—even files not created by WRITE-ON—and spool text to disk for letter printing or editing. Chain up to 100 files in a single printer run. Needs Applesoft and 32K.
On Disk with operating manual \$99.50

FILEMASTER I Here's a powerful data file manager giving you two programs—FORMAT & RETRIEVAL. It handles everything from phone lists to legal abstracts. You can design your own data structure with up to 500 characters per record and up to 15 searchable fields in any combination. Needs 32K. On Disk \$49.95

FILEMASTER II Has all the same features of FILEMASTER I plus allows for totaling, advanced math routines, more powerful print formatting, larger data fields, and disk-to-disk transfers \$99.50

+Space (Edu-Ware) Six programs form a unique epic game series. Multi-faceted simulation of life in interstellar society. You and opponents must make life & death decisions. Keeps track of your progress from one game to next. Needs 48K and Applesoft ROM. Disk \$29.95

Put It On! or our **Put It On!** at U. Gold II A collection of 49 programs for 16K Apple. Everything from Logic to action games. Only a buck a game. Specify I or II.
Price each: Tape \$49 Disk \$54

+Adventure This original, full-function game is the same as the one developed for large mainframes. Fight off pirates and vicious dwarfs. 700 travel options, 140 locations, 64 objects. Needs Applesoft ROM & 48K. Disk \$29.95

32K Disk Inventory Use stock numbers, description, vendor, record of purchase and sales date, amount on hand, cost & sell price, total value. Holds up to 300 items.
Disk \$40
+With Bill of Materials: Disk \$50

32K Data Base Cross file for phone lists, bibliographies, recipes. Run up to 9 lines of 40 columns each. Search by item anywhere.
Disk \$20

24K Hi-Res. Simulation Conway's equations on 296x180 screen. A mathematical simulation to demo population growth with birth, death and survival as factors. Tape \$10

16K Advanced Casino 9 gambling games: Roulette, Blackjack, Craps, Horse Race, Yahtzee, Keno, Slot Machine, Poker, and Acey-Ducey. Needs 16K. Tape \$29.95 Disk \$34.95

16K Space Battle You in your space capsule battle against the computer's saucer ... in hi-res graphics. Tape \$12

16K Memory Test Diagnostic routine to check range of memory. Indicates faulty addresses, data in memory cell, and faulty data. Tape \$5

16K Application Music synthesis composes original Irish jigs. Enter your own music and save on tape or disk. Includes 3 Bach fugues.
Tape \$10

+48K Edu-Pack (Edu-Ware) This package combines COMPU-READ—five speed reading programs; three PERCEPTION games where random shapes and sizes must be matched; and STATISTICS for computing Mean, Variance, Standard Deviation, and much more! Needs Applesoft ROM. Disk \$39.95

+32K Sargon II (Hayden) Here's the program that came in third against the big machines (mainframes and maxis) at the 9th North American Computer Chess Championship and placed first in the European Microcomputer Chess Championships! Has seven levels of play with Levels 0 - 3 playing in tournament time. Need a challenge? This is it!!
Tape \$29.95 Disk \$34.95

16K Hi-Res Baseball (Programma International) This animated simulation of a major league baseball game is for two players. The scoreboard is in the lower left of screen with the "throw pointer" for directing a throw in the lower right corner. Written entirely in machine language, the action is quick and smooth, making it the finest simulation of its kind.
Tape \$15.95

OTHER SPECIAL ITEMS FOR YOUR APPLE II

VERSAWRITER This digitizer drawing board, complete with a powerful software package on disk, lets you create any picture in color with high-resolution graphics. It's ideal for mass graphics. You can trace, edit, save and recall what you draw. It's a simple-to-use system for students, artists, engineers and graphic programmers. Has an 8½"x11" working area. New applications added include: •Text Writer adds text to your pictures. You control size, color and direction of text; •Electronic Drawing lets you create schematics and includes commonly used symbols for transistors, OPAMPS, and FETS; •Distance/Area lets you compute distances on maps or area of any frame. Applesoft ROM and 32K required. \$249.95
Add \$5 (U.S.) or \$10 (Foreign) for shipping.

Apple Monitor Pooled Everything you wanted to know about the Apple Monitor but couldn't figure out. All the PEEKS, POKES, and CALLS explained in an easy-to-understand form, written in plain English.
Only \$9.95

Don't see what you've been looking for, here? Then write for our FREE SOFTWARE CATALOG. We're saving one just for you!

To order, add \$2 (U.S.); \$5 (foreign) for shipping. California residents add 6% sales tax. Sorry, we cannot ship to P.O. Boxes. VISA/MasterCHARGE and BANKAMERICARD Welcomed!



Golden Plaza Shopping Center, Dept. 1A
9719 Reseda Blvd., Northridge, CA 91324
Telephone: (213) 349-5560

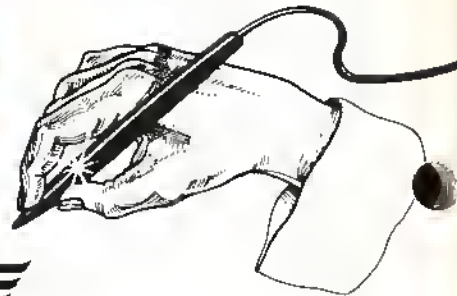


+ = Apple Plus compatible

What's NEW

from

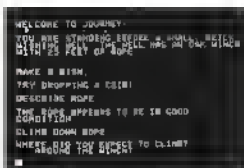
SOFTAPE



JOURNEY

You are about to embark on a very hazardous but profitable JOURNEY. The Apple is your eyes, ears, arms and legs. You can "GET" an object that is laying on the ground and you can travel North, East, South, West, Up and Down. You need to acquire tools as you JOURNEY forth and score precious points to become a GRANDMASTER JOURNEYER. JOURNEY REQUIRES 48K and loads on any Apple.

JAB-879 \$19.95



BASEBALL FEVER

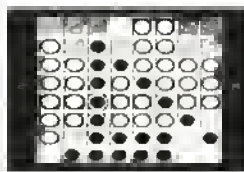
Catch the fever with this ball game that never gets rained out. You are the pitcher, batter and manager for your team. As manager you control line-ups, pitching staff and base stealers. The optional sound effects enhance the colorful animation. When the game is over you can save the game statistics to tape or disk and you are on your way to a winning season. Requires 32K or 48K with disk. INTEGER BASIC

BFM-879 \$12.95

OTHELLO

Play a true compelitor of this ancient game of territorial strategy. By flanking a line of the opponents men you "flip" them over to your own color. Be cautious though, for OTHELLO will never say die until the last move. OTHELLO is available for both INTEGER BASIC and APPLESOFT, and loads in 16K.

OHS-279 \$14.95



CONEY ISLAND

Enjoy the excitement of an amusement park at home. CONEY ISLAND has 22 varieties of paddle games that are fast. Written in FORTH II for speed, and using the beautiful color graphics of the Apple, one or two can play the most exciting paddle games yet written. CONEY ISLAND can be loaded on any Apple. 16K

CIW-879 \$12.95

FORTH II

FORTH II is an extremely well documented version of the Forth language that has been in use since the late 1960's.

It is many times faster than basic and is easy to use. Many of its features are as follows:

NEW FEATURES

- * RUNS ON ANY APPLE II COMPUTER
- * (24K minimum)
- * SUPPORTS DOS 3.2
- * CONTROL C BREAK AND CONTINUE
- * COMPATIBLE WITH AUTOSTART ROM
- * "SAVE IT" FILE FOR CUSTOMIZING SYSTEM

STANDARD FEATURES

- * INHERENTLY STRUCTURED LANGUAGE
- * DISK BASED EDITOR AND COMPILER
- * COMPLETE INSTRUCTIONAL REFERENCE MANUAL
- * EXCELLENT EXECUTION SPEED AND MEMORY EFFICIENCY
- * SUPPLIED ON MASTER DISKETTE
- * VERBS FOR GRAPHICS, GAME I/O, SOUND, DISK AND TAPE I/O

FOG-279 \$49.95

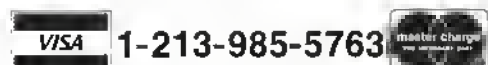
PADDLE PLUS

If you have the same problem as Arnold Zieback with constantly changing paddles and PENS, then you too need PADDLE PLUS. This extender plugs into your game I/O port and is conveniently secured for easy access.

PPA-180 \$14.95

WHERE TO GET IT: Look for the SOFTAPE Software display in your local computer store. Apple dealers throughout the United States, Canada, South America, Europe and Australia carry the SOFTAPE Software line of quality products.

If your local dealer is sold out of SOFTAPE Software you can order it direct from us by check or Visa/Master Charge. If you have any questions please call us at:



Or mail your order to the address below. We'll add your name to our mailing list for free literature and announcements of new products.

SOFTAPE

10432 Burbank Blvd. • North Hollywood, CA 91601

BRIGHT PEN

What is the difference between a light pen and the BRIGHT PEN. Intelligent software and extensive documentation. The software will help you to calibrate your system for optimum operation. The documentation details the BRIGHT PEN DRIVERS and how they are appended to your INTEGER BASIC programs. BRIGHT PEN Includes documentation booklet, two cassettes and, of course, the BRIGHT PEN.

BPE-279 \$34.95

DUMP-RESTORE

With DUMP-RESTORE you will be able to backup your disk files to cassette and restore them. This allows you to relocate disk space for maximum efficiency and speed. The programs are saved and restored individually or the entire disk can be saved and restored. DUMP-RESTORE loads with INTEGER BASIC and requires 32K.

DRG-879 \$14.95

RESET GUARD

Tired of hitting reset by mistake? If so RESET GUARD will solve the problem. RESET GUARD is a hardware package that plugs directly into your Apple. It protects your programs because it will only Reset if hit twice in one second. Guard your Apple and your sanity with RESET GUARD.

RGA-180 \$34.95

